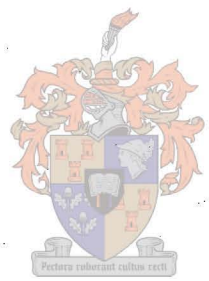
Adaptive Estimation of Speech Parameters

by

J.A.L. Basson



Study project submitted to the Faculty of Engineering, University of Stellenbosch, in partial fulfilment of the requirements for the degree of Master of Electronic Engineering.

Study leader: J.A. Du Preez

Stellenbosch, March 1994

Stellenbosch University https://scholar.sun.ac.za

Table of Contents

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

DECLARATION

I, the undersigned, hereby declare that the work contained in this study is my own original work and has not previously in its entirety or in part been submitted to any university for a degree.

Date: 22/2/94

Stellenbosch University https://scholar.sun.ac.za

Abstract

Linear predictive coding (LPC), and transformations of it, is currently the most popular way of analysing speech signals. Major limitations of using a frame-based technique are that each frame is analysed in isolation of the rest while assuming the excitation source to be a white, gaussian process. In order to reduce computation time, an all pole model is usually employed.

In this project an adaptive algorithm is proposed for speech signal analysis. The algorithm is based on the recursive least mean squares method with a variable forgetting factor. A pole-zero model is used to estimate the anti-formants present in certain sounds (i.e. nasals and nasalized vowels). This method offers better detection of poles and zeros in stationary environments and faster tracking of pole and zero frequencies in nonstationary signals than other sequential methods. An effective input estimation algorithm eliminates the influence of pitch on the parameter estimates by assuming the input to be a white noise process or a pulse sequence.

Stellenbosch University https://scholar.sun.ac.za

Opsomming

Linieêre voorspellings-kodering, en transformasies daarvan, is huidiglik die gewildste metode t.o.v. die analise van spraakseine. Blok-gebaseerde algoritmes het ernstige tekortkominge. Elke raam word byvoorbeeld in isolasie van die res geanaliseer terwyl daar aangeneem word dat die intree na die spraakkanaal 'n wit, gaussiese ruisproses is. Om berekeningstyd te beperk word 'n model met slegs pole gebruik.

In hierdie projek word 'n aanpasbare algoritme (gebaseer op die rekursiewe kleinste kwadrate metode) met 'n variërende vergeetfaktor voorgestel. 'n Pool-zero model bied akkurater opsporing van pole en zeros in stasionêre omgewings. Dit bied ook vinniger volging van pool en zero frekwensies in nie-stasionêre seine as ander aanpasbare algoritmes. 'n Effektiewe intree-beramings algoritme skakel die invloed van die fundamentele frekwensie op die beraamde parameters uit. Dit word reggekry deur te aanvaar dat die intree 'n wit ruis-proses óf 'n pols reeks kan wees.

Stellenbosch University https://scholar.sun.ac.za

Acknowledgements

I wish to express my deep appreciation and gratitude to Johan A. du Preez for his skilful guidance. His expertise was of invaluable assistance.

George W. Whitehead for providing space on the network drives and backups of the database.

My friend and colleague Jacques C. de Bruin for his moral support and assistance.

My parents, Liesl and Dakka for their love, patience, encouragement and financial support.

Stellenbosch University https://scholar.sun.ac.za

To God, His Son Jesus, and His Holy Spirit.

Who showed me the way.

Stellenbosch University https://scholar.sun.ac.za

Abbreviations

AIC	Akaike information criterion.
AR	autoregressive.
ARMA	autoregressive moving average.
CAT	criterion autoregressive transfer function.
CRLS	conventional recursive least squares.
FF	forgetting factor.
FIR	finite impulse response.
FPE	final prediction error.
FRLS	fractal recursive least squares.
HMM	hidden Markov model.
IIR	infinite impulse response.
LMS	least mean squares.
LPC	linear predictive coding.
LS	least squares.
LSL	least squares ladder.
MA	moving average.
MDL	minimum description length.
MRLS	modified recursive least squares.
MYWE	modified Yule Walker equations.
RLS	recursive least squares.
SEARMA	simultaneous estimation autoregressive moving average.
VFF	variable forgetting factor.
WRLS	weighted recursive least squares.
WRLS-VFF	Weighted Recursive Least Squares with a Variable Forgetting Factor.

Stellenbosch University https://scholar.sun.ac.za

List of Figures

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

List of Tables

Stellenbosch University https://scholar.sun.ac.za

List of Symbols

k	Time index.
y_k	Output of a digital filter at time k.
u_k	Input of a digital filter at time k.
a_k	AR parameter vector at time k.
b_k	MA parameter vector at time k.
$a_k(i)$	i'th element of the AR parameter vector at time k.
$b_k(i)$	i'th element of the MA parameter vector at time k.
p	AR order.
q	MA order.
σ_u^2	Variance of the signal 'u'.
m_u	Mean of the signal 'u'.
$H_k(z)$	Transfer function of a filter at time k.
$B_k(z)$	Numerator of the transfer function of a filter at time k.
$A_k(z)$	Denominator of the transfer function of a filter at time k.
$\mathcal{E}\{y_k\}$	The expected value of the sequence y_k.
ϕ_{yy}	Autocorrelation of the process y.
r'_{yy}	Unbiased *estimate* of the autocorrelation of the process y.
r_{yy}	Biased *estimate* of the autocorrelation of the process y.
r_{uy}	Cross-correlation between u and y.
y_k^f and y_k^b	Forward and backward estimates of y_k at time k.
e_k^f and e_k^b	Forward and backward linear prediction errors at time k.
a_k^f and a_k^b	Forward and backward prediction coefficients at time k.
$J(w)$	Mean square error criterion (probabilistic).
e_k	Estimation error.
$E(w,k)$, $E_k(\theta)$	Time-averaged mean square error criterion (deterministic).
$\hat{\theta}_k$	Estimated AR or ARMA parameter vector.
θ_k	Real AR or ARMA parameter vector.
ϕ_k	Data vector.
λ_k	Forgetting vector at time k.
Φ_k	Time-averaged covariance matrix at time k.
Θ_k	Time-averaged cross-covariance vector at time k.
P_k	Inverse of the covariance matrix of the RLS method.

Stellenbosch University https://scholar.sun.ac.za

K_k	Gain vector of the RLS method.
α_k	Innovations sequence or *a priori* estimation error.
$E_k(\theta)$	The error information or sum of the squared errors over the interval of interest.
u_k^w	White noise input signal.
u_k^p	Pulse input signal.
Ψ_k	Cross-covariance vector in the RLS method due to a pulse input.
λ_0	Lower threshold value for λ_k.
λ_{min}	Minimum value of λ_k.
P_0	Initial value of the inverse of the covariance matrix in the RLS method.
$\hat{\theta}_0$	Initial estimation of the AR or ARMA parameter vector.
E_0	The initial value for the error information or sum of the squared errors.
$\hat{u}_k$	Estimated input signal.
G_k	Gain of the filter.
C_n	The n'th cepstra coefficient.
α	Warping constant for the Mel-scale parameters.
K_m	The n'th reflection coefficient.
FD_k	Fractal dimension at time instant "k".
F_l	Lower thresshold of the fractal dimension.
F_z and F_p	Frequency of a zero or pole respectively.
B_z and B_p	Bandwidth of a zero or pole respectively.
E_{rms}	Root mean square error between the estimated and the original spectrograms.
E_p and E_q	Frequency estimation error of the poles and zeros respectively.

Stellenbosch University https://scholar.sun.ac.za

Chapter 1

Introduction

Some decades ago computer scientists and engineers began the task of writing computer programs that would convert speech sounds into written language, i.e., computer programs that would take dictation. This task has considerable practical importance since it will enable computers to "hear" for deaf individuals. Other applications include: automated bank transactions (like using a telephone to communicate with an automate) and auto-dialling facilities on telephones (i.e., you can order the phone to dial Peter's number by saying: "phone Peter").

Now, after more than three decades of work, even the most powerful super computer is still far less than adequate at what humans find to be a boringly simple task. The difficulty in solving pattern recognition problems, such as listening and understanding, is to figure out exactly how people distinguish relevant from irrelevant features of a pattern [3].

Once again then, attempts at computer simulation of a human ability led us, not to lessen the stature of Man, but to wonder anew at the complexity of the tasks that he performs apparently effortlessly.

1.1. MOTIVATION FOR THIS PROJECT

Speech-, language-, and speaker recognition are very important subjects in the field of human/machine interaction. In almost all applications the original speech signal has to be transformed before a machine interpretation. The first phase is the so-called signal processing phase. During this phase all the information in the signal is extracted and represented in a compact form. The next step is to classify the available data by using pattern recognition algorithms. It is important to realise that even the most powerful pattern recognition algorithm cannot compensate for inaccuracies in the initial signal processing phase. This investigation is aimed at producing a better model for this initial signal processing phase.

Stellenbosch University https://scholar.sun.ac.za

At the moment the most popular way to address the problem of extracting the information from a speech signal is to use spectrum analysis with an all-pole (Auto Regressive or AR) filter model. This method, also called a block analysis method, divides the speech signal into segments/frames of uniform lengths of approximately 20ms. An autoregressive (AR) filter model is fitted onto each of the frames. Thus, only one set of coefficients is obtained for each frame. If placement of the analysis window is not pitch synchronized (as in the case of our current system), the linear predictive coding (LPC) coefficients can vary, even during stationary speech. This leads to warbling in LPC speech. The reason for this is that in LPC analysis, the excitation source is assumed to be a white, gaussian process. Although this is a reasonable assumption for unvoiced sounds, it is definitely not the case for voiced speech [49] when the pitch pulse or pulses occur at random positions inside the analysis window. In the latter case a periodic pulse sequence would be more appropriate.

To minimize computational complexity, an all pole model is assumed for the speech source. This is a simplification of the actual vocal tract model, because continuous speech spectra contain zeros due to the glottal closure, as well as zeros from the vocal tract response. The existence of zeros in nasals and unvoiced sounds shows prominently on spectrograms. An all pole-model can only approximate zero positions. Furthermore, the presence of zeros causes the pole estimates to deviate from the actual formant values [49].

The resultant filter coefficients are used to generate formants or cepstra that are then representative of the original speech signal.

A major limitation of the above method is that each frame is initialized in isolation from the rest. This implies that the context in which a frame appears, relative to the surrounding frames, is disregarded.

An adaptive approach

By using adaptive methods to estimate the vocal tract model we hope to address the above problem. A better estimation of this model, if successful, can affect the results of all the pattern recognition techniques that follow this initial processing.

The main goal of the project is to eliminate the problems encountered in the above-mentioned block segmentation approach. The basic idea is to obtain a time-varying model that is unaffected by pitch pulse locations. The inclusion of zeros in the current all-pole model will also be investigated. An effective ARMA model could improve recognition of both nasal and unvoiced sounds.

Stellenbosch University https://scholar.sun.ac.za

1.2. REVIEW OF SPEECH ANALYSIS METHODS

Block Methods

The accurate estimation and tracking of pole and zero frequencies and their bandwidths has long been recognised as important subjects in both speech recognition and speech synthesis. Most parametric estimation algorithms assume separate models for the excitation and the vocal tract response. The vocal tract is usually modelled by analysing the speech with a linear predictive coding (LPC) technique, as discussed in the previous section. An estimate of the glottal excitation waveform can then be obtained by inverse filtering.

Several frame-based LPC techniques exist. Some of these include the (1) Yule Walker method, (2) Burg's maximum entropy method, (3) least squares autocorrelation method, (4) the covariance method and the (5) modified covariance method. A short discussion and summary of each of these is given in chapter 2. We will only mention two of the most important methods here.

In 1975 the most popular AR parameter estimation method was the maximum entropy method developed by Burg [45]. It provides increased spectral resolution over the earlier Yule Walker and autocorrelation methods while guaranteeing a stable filter. Two main disadvantages are that it suffers from spectral line splitting and a bias in the location of spectral peaks.

One year later in 1976, Ulrych and Clayton [42] and Nutall [43] proposed the modified covariance method. Although their method showed no apparent line splitting and less bias in spectral peaks, it was not used extensively due to the computational complexity of the order p^3 multiplications and additions (p= the AR order) per frame. In 1980, Marple [18] succeeded in reducing the computational complexity to more or less the same as that of the Burg algorithm (of the order of p^2 multiplications and additions per frame). The result was that the Marple algorithm has since become one of the most popular AR block-based algorithms [50], [51].

All of the AR block methods have two serious limitations because of assumptions made in their deductions:

- The speech is assumed to be stationary over the interval of interest. Thus, only one set of coefficients is obtained for each frame. If the frame were to be positioned over a non stationary part of the speech signal, the parameters for that frame would be inaccurate.

- In LPC analysis, the excitation source is assumed to be a white, gaussian process. Although this is a reasonable assumption for unvoiced sounds, it is definitely not the case for voiced speech [49] when a pitch pulse occurs in the

Stellenbosch University https://scholar.sun.ac.za

middle of the analysis window. In the latter case a periodic pulse train would be more appropriate.

We can limit the restriction of using a white gaussian noise process, as the input signal to the vocal tract, by placing the analysis window *between* pitch pulses. This technique is known as pitch synchronized LPC and usually requires an additional algorithm to locate the pitch pulses accurately. For this reason it is rarely used.

Thus several factors influence the accuracy of the parameters estimated with an AR-LPC method:

1. The placement of the analysis window.
2. The length of the analysis window.
3. The influence of the fundamental frequency, especially when it lies close to the first formant.
4. Spectral valleys due to anti-formants in nasal sounds [49] cause the formant estimates to deviate from their actual values.
5. Rapid changes in the formant positions occur at some vowel/consonant transitions, which cannot be followed by LPC methods.

Sequential methods offer many advantages over traditional frame-based methods, since they overcome most of the problems mentioned. A discussion of a few sequential methods that have been applied to speech analysis will follow.

Sequential Methods

As the name suggests, sequential parameter estimation techniques provide parameter estimates sequentially with time. Sequential methods do not have to wait until all the data points are collected, but update the parameter estimates as each new data point is received. Unlike frame-based techniques, adaptive algorithms can operate satisfactorily in non-stationary environments [56].

Several researchers proposed adaptive ARMA algorithms for estimating speech parameters. Most of the more successful adaptive algorithms are derived from the recursive least squares (RLS) algorithm, as discussed by Haykin [56]. Other sequential algorithms, like the least mean squares (LMS) and the Kalman methods are summarized in chapter 2. We shall now discuss a few sequential algorithms that have been applied to speech analysis.

In 1982 Morikawa and Fujisaki [7] proposed the sequential estimation ARMA (SEARMA) algorithm. Their method is based on the RLS method (see Haykin [56]). The SEARMA algorithm uses the estimation error as an estimate of the input signal to the vocal tract, which is assumed to be a white noise process. This assumption does not hold for voiced speech where the input signal can be a periodic pulse sequence. If the influence of the pitch pulses is not removed from the estimated parameters, when voiced speech is analysed, it will

Stellenbosch University https://scholar.sun.ac.za

decrease their accuracy [8], [11] ,[14]. A further shortcoming of the SEARMA method is that the forgetting factor is kept constant and equal to unity. This makes the SEARMA method only applicable to stationary processes.

Miyanaga *et al.* [11] proposed a method to estimate both white noise and pulses for the input signal. They used two adaptive algorithms to do this. This increased computational complexity dramatically. The first algorithm is used to estimate the parameters and compute the optimal estimation error and its covariance. The second algorithm is used to compute the actual estimation error and, by using the covariance of the optimal estimation error, decides whether the input is white noise or a pulse. In a later paper [12] they extended their method to include estimation of non-stationary parameters.

Ting and Childers [14] realized that it was computationally more efficient to update the forgetting factor by using the already available estimation error. They designed the weighted recursive least squares algorithm with a variable forgetting factor (WRLS-VFF) to estimate the ARMA parameters of the vocal tract. An effective input estimation algorithm uses the variable forgetting factor (VFF) to decide on white noise and pulse excitation. Thus only one, instead of two, adaptive process is required.

We may therefore summarise the advantages of using a sequential algorithm such as the WRLS-VFF, instead of a block approach, as follows:

1. The WRLS-VFF can accurately estimate and track both formant and anti-formant frequencies and their bandwidths.
2. The limitations of using an analysis window of fixed length are removed by employing a variable forgetting factor. The forgetting factor automatically shortens or increases the effective memory of the algorithm by using the available estimation error.
3. The influence of the fundamental frequency on the parameter estimates is eliminated with the use of an effective input estimation algorithm.
4. Spectral valleys due to anti-formants in nasal and some fricative sounds can be modelled by the zeros in the pole-zero estimation model.
5. A slight modification to the WRLS-VFF algorithm allows it to follow rapid changes associated with some vowel/consonant transitions.

1.3. PREVIEW OF THE CHAPTERS

In chapter 2 existing parameter estimation methods are discussed. Both block data algorithms as well as sequential algorithms are considered.

Chapter 3 consists mainly of a detailed description of the weighted recursive least squares algorithm with a variable forgetting factor (WRLS-VFF), developed by Ting and Childers [14]. The WRLS-VFF is applied to the estimation of a pole-zero

Stellenbosch University https://scholar.sun.ac.za

model for the vocal tract.

Chapter 4 provides the reader with practical procedures for the implementation of the WRLS-VFF algorithm applied to the identification of a pole-zero model for the vocal tract. Ways of selecting the autoregressive moving average (ARMA) orders, initiation of the WRLS-VFF algorithm and faster tracking of certain non-stationary signals are discussed.

A modified version of the WRLS-VFF is tested on various synthetic speech signals in chapter 5. Its performance is compared to that of 4 other sequential algorithms.

In chapter 6 we derive the relationship between ARMA parameters and ARMA cepstra as well as the relationship between ARMA parameters and Mel-scale cepstra.

A comparison between the proposed sequential algorithm and a popular frame-based method is given in chapter 7. The performance of both methods is tested on isolated word recognition.

Conclusions and recommendations for future research are given in chapter 8.

Stellenbosch University https://scholar.sun.ac.za

Chapter 2

Parameter Estimation Methods

2.1. INTRODUCTION

In this chapter we give a brief background of different parameter estimation methods. We will only consider the special class of models; those driven by white noise processes and possessing rational system functions. The output signal of such a system is uniquely described by the parameters of the model and the variance of the white noise process. This special class includes autoregressive (AR), moving average (MA) and autoregressive-moving average (ARMA) models.

In §2.2 we define the differential equations and transfer functions of the different models (MA, AR, ARMA). These definitions are used consistently throughout the rest of the text.

Algorithms that estimate the parameters of these models can be divided into two categories, namely: those that work on blocks of data and those that process data as it becomes available. Various block methods are summarised at the end of §2.3. Advantages and disadvantages of each method are pointed out. The first part of §2.3 is devoted to AR analysis, and specifically the derivation of the ARMA Yule Walker equations. The relationship between AR analysis and linear prediction, which is exploited by several frame based algorithms as well as sequential algorithms, is considered. In §2.4 we look at the basic idea of sequential processing. The adaptive filter is introduced in two adaptive structures, (1) the adaptive linear predictor and (2) the adaptive system identifier. Although no specific sequential algorithm is derived, a general cost function is defined. Minimizing this cost function leads us to the principal of orthogonality.

2.2. AR, MA AND ARMA MODELS

The ARMA model may be expressed, in linear difference equation form, as follows:

$$y_k = -\sum_{i=1}^{p} a_k(i) y_{k-i} + \sum_{i=0}^{q} b_k(i) u_{k-i} \qquad \text{with } a_k(0) = 1 \qquad 2.1$$

Stellenbosch University https://scholar.sun.ac.za

OR

$$y_k = \sum_{i=0}^{\infty} h_k(i) u_{k-i} \qquad 2.2$$

where:
k is a time index,
u_k is the input sequence and is a zero mean, unit variance white noise sequence,
h_k is the impulse response sequence at time k,
y_k is the output sequence of the causal ARMA system at time k,
a_k is the AR parameter sequence at time k, with order p.
b_k is the MA parameter sequence at time k, with order q.

For consistancy with later chapters we assume that $b_k(0) = 1$ and $a_k(0) = 1$ so that:

$$y_k = -\sum_{i=1}^{p} a_k(i) y_{k-i} + \sum_{i=1}^{q} b_k(i) u_{k-i} + u_k \qquad 2.3$$

This is done without any loss of generality because the filter input u_k can always be scaled to account for any filter gain. Note however that in such a case the sequence u_k will not have a unit variance anymore, but a variance of σ_u^2.

Figure 2.1 shows the input-output relationship of equation 2.3.

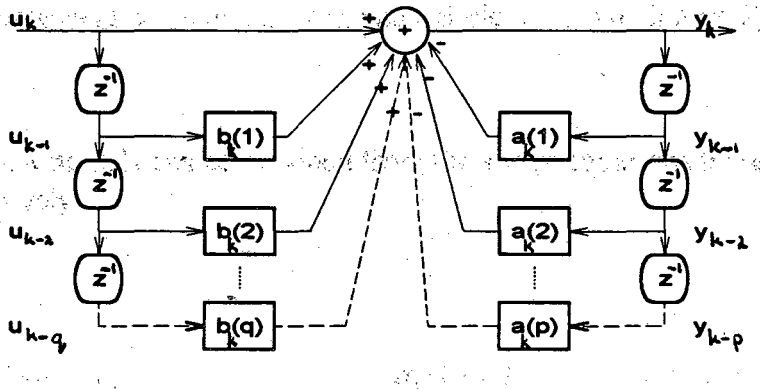


Figure 2.1 The input/output relationship of an ARMA Filter.

The transfer function of the ARMA process is expressed in rational form, as a polynomial of z^{-1}, as follows:

Stellenbosch University https://scholar.sun.ac.za

$$H_k(z) = \frac{B_k(z)}{A_k(z)} = \frac{\sum_{i=0}^{q} b_k(i) z^{-i}}{\sum_{i=0}^{p} a_k(i) z^{-i}} \qquad 2.4$$

$$H_k(z) = \frac{1 + \sum_{i=1}^{q} b_k(i) z^{-i}}{1 + \sum_{i=1}^{p} a_k(i) z^{-i}} \qquad b_k(0) = 1 \text{ and } a_k(0) = 1 \quad 2.5$$

The factored form of H(z) is then:

$$H_k(z) = \frac{b_k(0) \prod_{i=1}^{q} (1 - Z_k(i) z^{-i})}{\prod_{i=1}^{p} (1 - P_k(i) z^{-i})} \qquad 2.6$$

with Z_k (the roots of the numerator polynomial) the zeros of H(z) and P_k (the roots of the denominator polynomial) the poles of H(z).

The ARMA system is said to be stable if the magnitudes of each of the poles are less than unity or $|P_k| < 1$. A minimum phase linear system is a causal system in which all the poles and zeros lie inside the unit circle in the z-plane or $|P_k| < 1$ and $|Z_k| < 1$.

The AR and MA models can be obtained from the above equations by setting q=0 or p=0 respectively.

AR-model:

$$y_k = -\sum_{i=1}^{p} a_k(i) y_{k-i} + u_k \qquad a_k(0) = 1 \qquad 2.7$$

The input and output relations of this equation are shown in figure 2.2.

UNIVERSITEITSBIBLIOTEEK

Stellenbosch University https://scholar.sun.ac.za

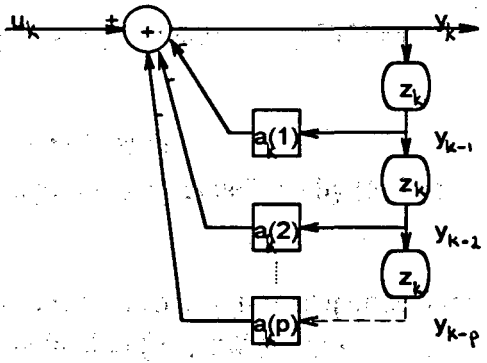


Figure 2.2 The input/output relationship of an AR Filter.

MA-model

$$y_k = \sum_{i=1}^{q} b_k(i)u_{k-i} + u_k \qquad b_k(0) = 1 \qquad 2.8$$

The input and output relations of this equation are shown in figure 2.3.

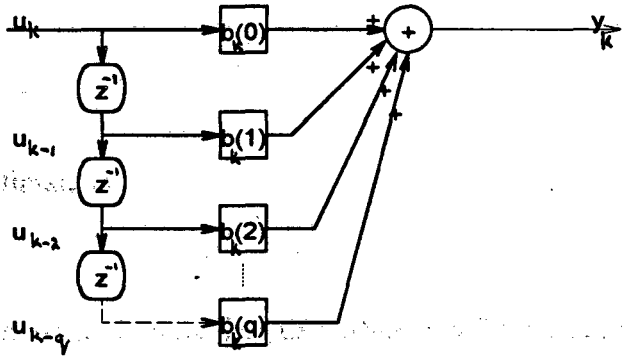


Figure 2.3 The input/output relationship of an MA Filter.

2.3. BLOCK METHODS

We divide the block techniques into three subclasses, based on how the parameters are obtained. Algorithms that estimate the model parameters by using an estimate of the *autocorrelation* sequence include the Yule Walker and Autocorrelation methods. The Burg algorithm estimates the model parameters through the equivalent representation of *reflection coefficients*. Two algorithms, the covariance and modified-covariance methods, estimate the AR parameters directly using a least squares method.

2.3.1. Autocorrelation Estimation

An accurate way of estimating the autocorrelation is of great importance for many of the AR analysis techniques that will be discussed later. A review of two commonly used estimations is given.

Stellenbosch University https://scholar.sun.ac.za

The mean of a random process y_k is defined by [55] as :

$$m_y = \mathcal{E}\{y_k\} \qquad 2.9$$

where $\mathcal{E}$ denotes the expected value.
The variance of a random process y_k is defined by [55] as:

$$\sigma_y^2 = \mathcal{E}\{(y_k - m_y)^2\} \qquad 2.10$$

The autocorrelation of a random process y_k is dependent on the time difference between samples and is defined in [51] and [55] as :

$$\phi_{yy}(m) = \mathcal{E}\{y_k y_{k+m}^*\} \qquad 2.11$$

The unbiased autocorrelation estimate is given by [55] as:

$$r'_{yy}(k) = \frac{1}{N-|m|}\sum_{k=0}^{N-|m|-1} y_k y_{k+m}^* \qquad \text{with } |m| < N \qquad 2.12$$

The biased autocorrelation estimate is given by [55] as:

$$r_{yy}(k) = \frac{1}{N}\sum_{k=0}^{N-|m|-1} y_k y_{k+m}^* \qquad 2.13$$

The bias of this estimate is:

$$bias = \frac{m}{N}\phi_{yy}(m) \qquad 2.14$$

As $N \to \infty$ the bias becomes less, thus this estimation for the autocorrelation function is asymptotically unbiased.

2.3.2. Analysis

A method to solve the ARMA parameters of equation 2.1 is presented here [47]. The solution of the AR parameters of a purely AR system is a special case of this solution, and can be found at the end of this section.

If we multiply equation 2.1 on both sides with y_{k-m}^* and take the expectation we get:

$$\mathcal{E}\{y_k y_{k-m}^*\} = -\sum_{i=1}^{p} a_k(i)\mathcal{E}\{y_{k-i}y_{k-m}^*\} + \sum_{i=0}^{q} b_k(i)\mathcal{E}\{u_{k-i}y_{k-m}^*\} \qquad 2.15$$

$$\text{OR} \quad r_{y_k y_k}(m) = -\sum_{i=1}^{p} a_k(i) r_{y_k y_k}(m-i) + \sum_{i=0}^{q} b_k(i) r_{u_k y_k}(m-i) \qquad 2.16$$

where:

$r_{y_k y_k}(m)$ is the autocorrelation of the output series y_k,

Stellenbosch University https://scholar.sun.ac.za

$r_{u_k y_k}(m)$ is the cross correlation between the input u_k and the output y_k.

Equation 2.16 is the classical ARMA Yule Walker Normal Equation [47]. This equation is sometimes referred to as the Modified Yule Walker Equation (MYWE).

Pre-multiply equation 2.2 on both sides with u_{k-m} and take the expectation. The cross correlation $r_{u_k y_k}(m)$ can be expressed in the following format:

$$r_{u_k y_k}(m) = r_{u_k u_k}(m) + \sum_{i=1}^{\infty} h_k(i) r_{u_k u_k}(m-i) \qquad 2.17$$

with u_k a white noise process, thus $r_{u_k u_k}(m) = 0$ for $m > 0$:

$$r_{u_k y_k}(m) = \begin{cases} 0 & m > 0 \\ \sigma_{uu}^2 & m = 0 \\ \sigma_{uu}^2 h_k(-m) & m < 0 \end{cases} \qquad 2.18$$

Substitute equation 2.18 back into 2.16 to obtain:

$$r_{y_k y_k}(m) = \begin{cases} -\sum_{i=1}^{p} a_k(i) r_{y_k y_k}(m-i) + \sigma_{uu}^2 \sum_{i=m}^{q} b_k(i) h(i-m) & 0 \le m \le q \\ -\sum_{i=1}^{p} a_k(i) r_{y_k y_k}(m-i) & m > q \\ r_{y_k y_k}^{*}(-m) & m < 0 \end{cases} \qquad 2.19$$

The MA parameters in equation 2.19 are convolved with the impulse response. In order to solve the MA parameters, a nonlinear set of equations has to be evaluated. We can however compute the AR parameters from equation 2.19 by evaluating it for $m > q$. The matrix representation for $q+1 \le m \le q+p$ follows:

Stellenbosch University https://scholar.sun.ac.za

$$-\begin{bmatrix} r_{y_k y_k}(q) & r_{y_k y_k}(q-1) & \cdots & r_{y_k y_k}(q-p+1) \\ r_{y_k y_k}(q+1) & r_{y_k y_k}(q) & \cdots & r_{y_k y_k}(q-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{y_k y_k}(q+p-1) & r_{y_k y_k}(q+p-2) & \cdots & r_{y_k y_k}(q) \end{bmatrix} \begin{bmatrix} a_k(1) \\ a_k(2) \\ \vdots \\ a_k(p) \end{bmatrix} = \begin{bmatrix} r_{y_k y_k}(q+1) \\ r_{y_k y_k}(q+2) \\ \vdots \\ r_{y_k y_k}(q+p) \end{bmatrix}$$

2.20

For a purely AR system the AR Yule Walker equations from 2.19 (set q=0) are:

$$r_{y_k y_k}(m) = \begin{cases} -\sum_{i=1}^{p} a_k(i) r_{y_k y_k}(m-i) & m > 0 \\ -\sum_{i=1}^{p} a_k(i) r_{y_k y_k}(m-i) + \sigma_{uu}^2 & m = 0 \\ r_{y_k y_k}^*(-m) & m < 0 \end{cases} \quad 2.21$$

We can compute the AR parameters from equation 2.21 by evaluating it for $m \geq 0$. The matrix representation for $0 \leq m \leq p$ follows. In matrix form:

$$\begin{bmatrix} r_{y_k y_k}(0) & r_{y_k y_k}(-1) & \cdots & r_{y_k y_k}(-p) \\ r_{y_k y_k}(1) & r_{y_k y_k}(0) & \cdots & r_{y_k y_k}(-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ r_{y_k y_k}(p) & r_{y_k y_k}(p-1) & \cdots & r_{y_k y_k}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_k(1) \\ \vdots \\ a_k(p) \end{bmatrix} = \begin{bmatrix} \sigma_{uu}^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad 2.22$$

For an MA system the MA Yule Walker equations from 2.19 (set p=0) are:

$$r_{y_k y_k}(m) = \begin{cases} 0 & m > q \\ \sigma_{uu}^2 \sum_{i=m}^{q} b_k(i) h_k(i-m) & 0 \leq m \leq q \\ r_{y_k y_k}^*(-m) & m < 0 \end{cases} \quad 2.23$$

Stellenbosch University https://scholar.sun.ac.za

2.3.3. Linear Prediction

In this section we shall derive the relationship between AR analysis and linear prediction analysis according to Marple [50]. This relationship is exploited by several of the algorithms to be discussed in §2.3.5. First, we define the forward linear prediction estimate as a linear combination of past observations:

$$\hat{y}_k^f = -\sum_{i=1}^{p} a_k^f(i) y_{k-i} \qquad 2.24$$

where:

$\hat{y}_k^f$ is the forward estimate of y_k at time k,

a_k^f is the forward linear prediction coefficient sequence at time k.

The forward linear prediction error at time k is defined as:

$$e_k^f = y_k - \hat{y}_k^f \qquad 2.25$$

with variance:

$$\sigma_{e^f}^2 = \mathcal{E}\{e_k^f e_k^{f*}\} = \mathcal{E}\{|e_k^f|^2\} \qquad 2.26$$

We now substitute equations 2.24 and 2.25 into 2.26

$$\sigma_{e^f}^2 = r_{yy}(0) + \sum_{i=1}^{p} a_k^f(i) r_{yy}(-i) + \sum_{j=1}^{p} [a_k^f(j)]^* r_{yy}(j) + \sum_{i=1}^{p}\sum_{j=1}^{p} a_k^f(i)[a_k^f(i)]^* r_{yy}(j-i) \qquad 2.27$$

We know that $r_{yy}(-i) = r_{yy}^*(i)$ for the wide sense stationary process y_k. Equation 2.27 can be written in the following matrix form:

$$\begin{bmatrix} r_{yy}(0) & r_{yy}^*(1) & \cdots & r_{yy}^*(p) \\ r_{yy}(1) & r_{yy}(0) & \cdots & r_{yy}^*(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}(p) & r_{yy}(p-1) & \cdots & r_{yy}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_k^f(1) \\ \vdots \\ a_k^f(p) \end{bmatrix} = \begin{bmatrix} \sigma_{e_k^f}^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad 2.28$$

This matrix is the equivalent of the AR Yule Walker equations for the system:

$$y_k = -\sum_{i=0}^{p} a_k^f(i) y_{k-i} + e_k^f \qquad 2.29$$

Stellenbosch University https://scholar.sun.ac.za

There are two important differences between the AR Yule Walker equations and equation 2.29 namely:

e_k^f is the output (not the input !) of the forward linear prediction error filter and y_k is the input. In the AR Yule Walker system these two are interchanged.

e_k^f is uncorrelated with y_k, but is not a white noise process unless y_k was generated as an AR process with order p. Only then will e_k^f be a white noise process and $a_k^f = a_k$ (the real AR parameters). In this case the prediction filter will be a whitening filter.

So far, the equations in this section were deduced for the forward prediction error filter. The equations for the backward prediction error filter can be obtained by following almost the same procedure. First, we define the backward linear prediction estimate as a linear combination of future observations:

$$\hat{y}_k^b = -\sum_{i=0}^{p} a_k^b(i) y_{k+i} \qquad 2.30$$

where:

$\hat{y}_k^b$ is the backward estimate of y_k at time k,

a_k^b is the backward linear prediction coefficient sequence at time k.

The backward linear prediction error at time k is defined as:

$$e_k^b = y_{k-p} - \hat{y}_{k-p}^b \qquad 2.31$$

with variance:

$$\sigma_{e^b}^2 = \mathcal{E}\{e_k^b e_k^{b*}\} = \mathcal{E}\{|e_k^b|^2\} \qquad 2.32$$

We now substitute equations 2.30 and 2.31 into 2.32 to obtain:

$$\sigma_{e^b}^2 = r_{yy}(0) + \sum_{i=1}^{p} a_k^b(i) r_{yy}(i) + \sum_{j=1}^{p} [a_k^b(j)]^* r_{yy}(-j) + \sum_{i=1}^{p}\sum_{j=1}^{p} a_k^b(i)[a_k^b(i)]^* r_{yy}(j-i) \qquad 2.33$$

We know that $r_{yy}(-i) = r_{yy}^*(i)$ for the wide sense stationary process y_k. Equation 2.33 can be written in the following matrix form:

Stellenbosch University https://scholar.sun.ac.za

$$\begin{bmatrix} r_{yy}(0) & r_{yy}^*(1) & \cdots & r_{yy}^*(p) \\ r_{yy}(1) & r_{yy}(0) & \cdots & r_{yy}^*(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}(p) & r_{yy}(p-1) & \cdots & r_{yy}(0) \end{bmatrix} \begin{bmatrix} a_k^b(p) \\ \vdots \\ a_k^b(1) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \sigma_{e_k^b}^2 \end{bmatrix} \qquad 2.34$$

2.3.4. AR-Methods

Yule Walker

The easiest and most obvious way to determine the AR parameters for a series of data samples is to obtain an estimate of the autocorrelation by using equation 2.12 or 2.13 to solve the AR Yule Walker matrix in 2.22. This technique is rarely used, because it produces poor-resolution spectral estimates, especially when only short data segments are examined.

If the unbiased autocorrelation estimate is used, it may result in unstable filters, because the autocorrelation matrix is not guaranteed to be positive-definite.

Summary:

- Bias in the location of the spectral peaks.
- Spectral line splitting occurs when signal-to-noise ratio (SNR) is high, or the number of AR parameters is a large percentage of the number of samples in the data record.

Burg

Another name for this method is the Maximum Entropy Method (MEM). It was first introduced by Burg in 1975 [45].

Summary:

- This is a least squares method.
- The method estimates the reflection coefficients.
- The error power to be minimized is the arithmetic mean of the forward and backward linear prediction squared errors:

$$\sigma_{fb}^2 = \frac{1}{2(N-p)}\left[\sum_{i=p}^{N-1}\left|e_p^f(i)\right|^2 + \sum_{i=p}^{N-1}\left|e_p^b(i)\right|^2\right]$$

Stellenbosch University https://scholar.sun.ac.za

- Optimization with respect to a single parameter, namely the reflection coefficient k_p, is achieved by constraining the AR coefficients to satisfy the Levinson algorithm.
- The estimated filters are always stable.
- Bias in the location of the spectral peaks.
- Spectral line splitting occurs when SNR is high, or the number of AR parameters is a large percentage of the number of samples in the data record.

Autocorrelation Method

Kay [51] also refers to this as the Yule Walker method, because of its equivalence to the Yule Walker equations with the biased autocorrelation coefficients.

Summary:

- This is a least squares method.
- The error power to be minimized is:

$$\sigma_f^2 = \frac{1}{2(N+p)} \sum_{i=0}^{N+p-1} \left| e_p^f(i) \right|^2 \quad \text{and}$$

$$\sigma_b^2 = \frac{1}{2(N+p)} \sum_{i=0}^{N+p-1} \left| e_p^b(i) \right|^2$$

- It is sometimes called the autocorrelation method because the summation ranges in the error power are that of the windowed case.
- Separate minimizing of the forward and backward linear prediction squared errors is achieved with respect to all the prediction coefficients.
- The estimated filters are always stable.
- The parameters can be solved with the Levinson algorithm.
- It is rarely used for the same reason as for that of the AR Yule Walker method with the biased autocorrelation estimate.
- The windowing of this method is responsible for the reduction in spectral resolution relative to other methods.

Covariance Method

Summary:

- This is a least squares method.

Stellenbosch University https://scholar.sun.ac.za

- The error powers to be minimized are:

$$\sigma_f^2 = \frac{1}{2(N-p)} \sum_{i=p}^{N-1} \left| e_p^f(i) \right|^2 \quad \text{and}$$

$$\sigma_b^2 = \frac{1}{2(N-p)} \sum_{i=p}^{N-1} \left| e_p^b(i) \right|^2$$

- It is also called the covariance method because the summation ranges in the error power are those of the unwindowed case.
- Separate minimizing of the forward and backward linear prediction squared errors is achieved with respect to all the prediction coefficients.
- A stable filter is not guaranteed.

Modified Covariance Method

Ulrych and Clayton [42] and Nutall [43] proposed this method simultaneously and independently in 1976. Ulrych and Clayton named it the least squares method although there are many other least squares methods available to choose from. Nutall called his algorithm the forward-backward prediction method. Sometimes this method is also referred to as the Marple algorithm. Marple [18] reduced the computational complexity of the least squares forward/backward prediction method from NM^2 to NM. Where N= the amount of data samples and M= the order of the AR process. He obtained this by eliminating the matrix inversion through a recursive method. The Levinson constraint (see Burg) was also removed.

Thus the Marple algorithm is essentially a recursive algorithm, based on an unconstrained least squares solution for the AR parameters by using forward/backward linear prediction.

Summary:

- This is a least squares method.
- The error power to be minimized is:

$$\sigma_{fb}^2 = \frac{1}{2(N-p)} \left[\sum_{i=p}^{N-1} \left| e_p^f(i) \right|^2 + \sum_{i=p}^{N-1} \left| e_p^b(i) \right|^2 \right]$$

- It is also called the modified covariance method because the summation ranges in the error power is that of the unwindowed case.
- Combined minimizing of the forward and backward linear prediction squared errors is achieved with respect to all the prediction coefficients
- No apparent line splitting occurs when Marple's order determination technique is used.
- Less bias occurs in the spectral peaks than in Burg's method [45].

Stellenbosch University https://scholar.sun.ac.za

- For $p << N$ this method is more efficient than Burg's algorithm.
- It does not guarantee a stable filter.
- The modified covariance method always yields stable reflection coefficients.
- Stable lattice filters will result from this method.

2.4. SEQUENTIAL METHODS

Sequential techniques have many advantages in situations where large amounts of data need to be processed. These estimation methods are especially suited for tracking slowly time-varying signals. The AR or ARMA parameters are updated as each new data sample is received. It is not necessary to wait until all the data points are collected. These are also referred to as adaptive methods.

The basics of adaptive filter theory used for adaptive parameter estimation will be covered in §2.4.1. The sequential algorithms for adaptive AR estimation can be divided into two categories [50]. The first class is based on the Wiener filter approach, modified through the use of a gradient approximation method, and includes the Least Mean Squares (LMS) algorithm. The second class is derived from the classical Kalman algorithm and includes the Recursive Least Squares (RLS) algorithm. Properties of both of these will be discussed in the following paragraphs.

2.4.1. The Basic Adaptive Filter

The ability of adaptive filters to operate satisfactorily in a non-stationary environment makes them a powerful tool for signal processing applications. The basic structure of an adaptive filter is shown in figure 2.4.

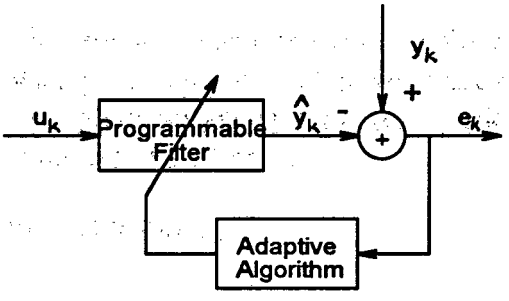


Figure 2.4 The Adaptive Filter.

Stellenbosch University https://scholar.sun.ac.za

The adaptive filter operates in two phases:

1. The input signal u_k is filtered by the programmable (adaptive) filter to yield the output $\hat{y}_k$ of the filter.
2. The output $\hat{y}_k$ is then compared to the desired output y_k to yield an error signal $e_k = y_k - \hat{y}_k$. This error is used by an adaptive algorithm to adjust the weights in the programmable filter so as to minimize the error in some sense.

The adaptive filter in figure 2.4 can be a finite impulse response (FIR) or infinite impulse response (IIR) filter or a combination of the two.
The choice of the adaptive algorithm (see figure 2.4) is determined by various factors described by Haykin [56]:

1. *Rate of convergence*: This is the number of iterations that is required for the algorithm to converge to the optimum Wiener solution in the mean square sense, in response to a stationary input signal.
2. *Misadjustment*: Haykin [56] defines this as follows. "For an algorithm of interest, this parameter provides a quantitative measure of the amount by which the final value of the mean-squared error , averaged over an ensemble of adaptive filters, deviates from the minimum mean-squared error produced by the Wiener filter."
3. *Tracking*: This is the ability of an adaptive filtering algorithm to follow or *track* the statistical variations in a signal. A compromise between rate of convergence and steady-state fluctuation, due to algorithm noise, has to be agreed on for the best possible tracking performance.
4. *Robustness*: A robust algorithm can operate satisfactorily with ill-conditioned input data. The input data is defined as ill conditioned when the condition number of the underlying correlation matrix is high.
5. *Computational complexity*: This includes the number of floating point operations (additions and multiplications) per iteration required by the algorithm.
6. *Structure*: The structure of the algorithm determines how it will be implemented in hardware form.

We shall be concerned primarily with using the adaptive filter for parameter estimation. In the following paragraphs two adaptive structures will be discussed. They are the adaptive linear predictor and system identification models. Both of these can be used for parameter estimation.

2.4.2. Adaptive Linear Prediction

An adaptive linear predictor, shown in figure 2.5, is used to predict the present value of a signal from past values of the same signal. The present value of the signal serves as the desired response of this filter.

Stellenbosch University https://scholar.sun.ac.za

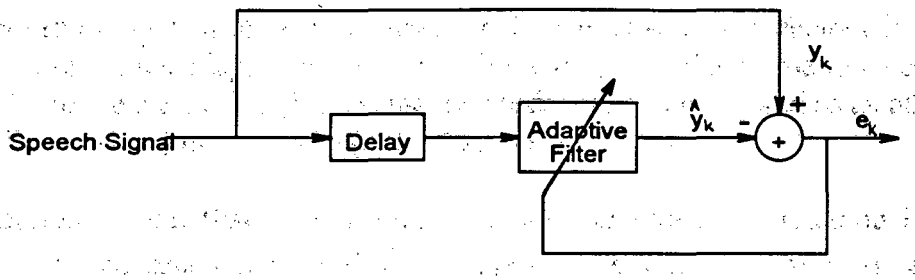


Figure 2.5 The Adaptive Linear Predictor.

Mathematically this filter can be expressed as:

$$\begin{aligned} e_k &= y_k - \hat{y}_k \\ &= y_k - \sum_{i=1}^{p} w_k(i) y_{k-i} \end{aligned} \qquad 2.35$$

Where $w_k(i)$ is the coefficient of the adaptive filter. If we assume that the error signal in figure 2.5 is equivalent to the input white noise u_k, of the AR model in figure 2.2, then the adaptive linear predictor can provide the same spectral representation as an AR model. At convergence the weighted coefficients, $w_k(i)$, of the adaptive linear predictor, will approach the AR coefficients, a_k.

2.4.3. Adaptive System Identification

An adaptive filter can be used to identify an unknown system as shown in figure 2.6.

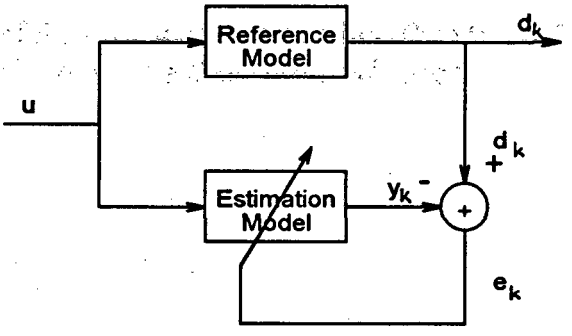


Figure 2.6 The Adaptive System Identification.

In speech modelling, the reference model in figure 2.6 can be considered as the vocal tract system. The estimated model (an adaptive filter) adjusts itself, causing

Stellenbosch University https://scholar.sun.ac.za

its output to follow that of the reference model. A least squares technique is generally used in the adaptive algorithm to fit the output of the estimation model to that of the reference model. The estimation model can be configured as an all-pole (AR), all-zero (MA) or pole-zero (ARMA) system.

It is important to note that both the reference model and the estimated model are supplied with the same input signal u_k. This poses a serious problem for speech processing, because the input signal to the vocal tract is not available! We shall discuss ways to overcome this obstacle in subsequent chapters.

2.4.4. The Cost Function

Defining a cost function is an integral part of every optimizing filter. Although there is no unique cost function for all applications, many popular functions exist to choose from. One of the most popular cost functions in use is the *mean-square error criterion*.

In deriving the Wiener equations (or Kalman approach), the *mean-square error criterion* is defined by Haykin [56] as follows:

$$J(w) = \mathcal{E}\left\{|e_k|^2\right\} \qquad 2.36$$

with

$$e_k = y_k - \hat{y}_k \qquad 2.37$$

The estimation error e_k is defined as the difference between the desired response (y_k) and the estimated response ($\hat{y}_k$). By using this error criterion, we follow a probabilistic approach, because the expectation operator denotes *ensemble averaging*[1].

In the development of the least squares algorithms (like the LMS and RLS) the cost function is described in terms of a *time-averaged*[2] error:

$$E(w, i_{1,2}) = \sum_{k=i_1}^{i_2} |e_k|^2 \qquad 2.38$$

with

$$e_k = y_k - \hat{y}_k \qquad 2.39$$

Note the important difference between the ensemble averaged- and the time averaged versions of the cost function, namely:

$J(w)$ is dependent only on the estimated parameters (w), while $E(w, i_{1,2})$ is

[1] An ensemble average is an average "across the realizations of the process" [56]
[2] A time averaged error is an average "along a realization of the process" [56]

Stellenbosch University https://scholar.sun.ac.za

dependent on both w and the observation interval ($i_{1,2}$). By introducing the time variable, k, into the cost function, we allow the tap-weight vector, w, to vary according to the observation interval. This comes in handy if the process under consideration is time-varying.

The type of data windowing employed is determined by the observation limits i_1 and i_2 in equation 2.38. Table 2.1 lists the different windowing methods with **p** the number of poles in the AR-model and **N** the total amount of observed data available.

Name of method	Lower limit i_1	Upper limit i_2	Comment
Covariance	p	N-1	No assumption is made about the data outside y_0 and y_{N-1}.
Autocorrelation	0	N-1+p	Data points prior to y_0 and after y_{N-1} equal zero.
Prewindowing	0	N-1	Data points prior to y_0 equal zero.
Postwindowing	p	N-1+p	Data points after y_{N-1} equals zero.

Table 2.1 The different windowing methods.

2.4.5. The Principle of Orthogonality

Let us define the estimation error in a certain adaptive system as:

$$e_k = y_k - \hat{y}_k \qquad k = i_1 \ldots i_2 \qquad 2.40$$

where

$$\hat{y}_k = -\sum_{i=1}^{p} a_k(i) y_{k-i} + \sum_{i=1}^{q} b_k(i) u_{k-i} \qquad k = i_1 \ldots i_2 \qquad 2.41$$

Equation 2.41 is closely related to equation 2.3, with p the poles of the AR system and q the zeros in the MA system. Here we assume that the input is totally unknown. Therefore, the signal y_k can be predicted only approximately from the linearly weighted summation of past samples. We can rewrite equation 2.41 in a

Stellenbosch University https://scholar.sun.ac.za

more comfortable form, using the following equivalent matrix notation:

$$\hat{y}_k = \phi_k^t \theta_k \qquad k = i_1 \ldots i_2 \qquad 2.42$$

where:

$$\theta_k^t = \begin{bmatrix} a_k(1) & a_k(2) & \cdots & a_k(p) & b_k(1) & b_k(2) & \cdots & b_k(q) \end{bmatrix} \qquad 2.43$$

$$\phi_k^t = \begin{bmatrix} -y_{k-1} & -y_{k-2} & \cdots & -y_{k-p} & u_{k-1} & u_{k-2} & \cdots & u_{k-q} \end{bmatrix} \qquad 2.44$$

The relationships in the above equations are shown in figure 2.7.

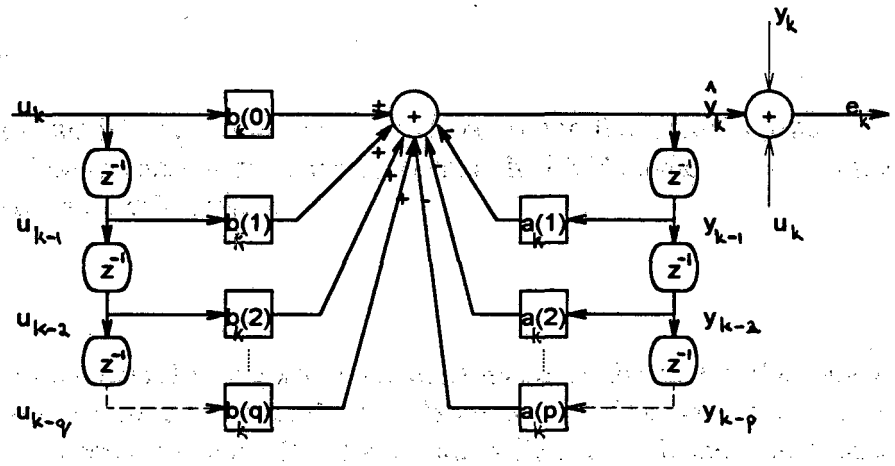


Figure 2.7 The adaptive ARMA estimation model.

Consider the cost function defined in equation 2.38, in the previous section, replacing the tap-weight vector with the newly defined tap weight vector, θ.

$$E(\theta, k) = \sum_{k=i_1}^{i_2} |e_k|^2 \qquad k = i_1 \ldots i_2 \qquad 2.45$$

To optimize the adaptive filter, we minimize $E(\theta, k)$ by differentiating with respect to θ. The procedure is shown in appendix C. The result is shown in the equation below:

$$\sum_{k=i_1}^{i_2} (e_k^*) \phi_k(i) = 0 \qquad i = 1 \ldots p+q \qquad 2.46$$

We interpret equation 2.46 as stating that, *for the adaptive filter to achieve its minimum least squares condition, each time series represented by the elements of*

Stellenbosch University https://scholar.sun.ac.za

the input vector ϕ_k, must be orthogonal to the estimation error e_k (see Haykin [56]). By making small changes to equation 2.46, it can be used as a basis for deducing all the least squares methods.

2.4.6. Adaptive AR Methods

LMS

The Least mean squares (LMS) algorithm is a stochastic gradient algorithm and is used for modelling stationary signals or very slow time-varying signals. Convergence is slow when compared to other sequential techniques. It is however simple to implement and computationally cheap. A major drawback of the LMS algorithm is that the rate of convergence is influenced by the eigenvalue spread (ratio of the maximum to the minimum eigenvalue) of the correlation matrix [56].

KALMAN

This is a recursive technique and is closely related to the recursive least squares method. Convergence is an order faster than the LMS method, but with the expense of more calculations.

LS

In the least squares (LS) algorithm we must minimize the sum of the squared errors. We also need to invert the correlation matrix. The weighting factor is constant and equal to unity. It is computationally extremely inefficient, because a matrix inversion (correlation matrix) is required at each time step. There is no recursion. The derivation is dealt with in the next chapter.

RLS

The recursive least squares (RLS) is the LS method, but modified to operate recursively. The correlation matrix is inverted by using the matrix inversion lemma. Conversion in an order faster than that of the LMS method. A forgetting factor is built into the RLS algorithm, which makes it especially useful for application to non-stationary situations. The derivation is also done in the next chapter. Misadjustment increases with a small forgetting factor, but it provides faster tracking of non-stationary signals. We shall develop the WRLS-VFF algorithm from the basic RLS method as discussed by Haykin [56].

2.5. CONCLUSION

In this chapter a brief summary of commonly used frame-based AR algorithms was presented. Over the years the Marple algorithm (Modified Covariance method)

Stellenbosch University https://scholar.sun.ac.za

proved to be a very effective and computationally efficient way of analysing speech. The current speech recognition system of the University of Stellenbosch is therefore based on this method.

Certain shortcomings of the frame based techniques (see Chapter 1) persuaded us to examine the viability of using sequential algorithms for analysing speech. From preparatory research we decided to use an RLS based algorithm. In the next chapter this algorithm will be developed.

Stellenbosch University https://scholar.sun.ac.za

Chapter 3

The Weighted Recursive Least Squares Algorithm with Variable Forgetting Factor (WRLS-VFF)

3.1. INTRODUCTION

In this chapter a detailed description of a WRLS-VFF (Ting and Childers [14]) is given, with emphasis on its application on estimating the ARMA vocal tract model.

In deriving the WRLS-VFF we shall start right at the beginning, with the unknown ARMA system. In order to solve the unknown parameters, the weighted recursive least squares (WRLS) algorithm will be deduced. During the course of this deduction we assume that the input to the vocal tract is available. The next step is to find a way to vary the forgetting factor according to the degree of stationarity detected in the speech signal. The chapter is concluded with an input estimation algorithm that allows a white noise or a pulse input signal.

3.2. WRLS ALGORITHM

Suppose the unknown vocal tract system can be modelled as an ARMA process, then the output sequence y_k can be generated according to the following equation:

$$y_k = -\sum_{i=1}^{p} a_k(i) y_{k-i} + \sum_{i=1}^{q} b_k(i) u_{k-i} + u_k \qquad k = 0 \ldots N-1 \qquad 3.1$$

where the input to the filter, u_k, is a zero mean white gaussian noise process, and a_k and b_k are the AR and MA parameters, respectively. The orders of the AR and MA processes are p and q respectively. Ways of determining the order will be discussed later. According to table 2.1, prewindowing is assumed, because all data before $k = 0$ and after $k = N-1$ is assumed to be zero.

It is noted that the output of the filter is dependent on the input signal, u_k.

Stellenbosch University https://scholar.sun.ac.za

Unfortunately this input signal is not available to us in speech processing. In order to provide accurate estimates of the ARMA parameters, the intended algorithm will have to include a reliable input estimation algorithm. Such an algorithm was developed by Ting and Childers [14] and will be discussed at a later stage. For now, we assume that the estimate of u_k is a known quantity at instant k. The estimated value of u_k will be called $\hat{u}_k$. With the parameter vector (θ_k) and the data vector (ϕ_k) defined as in 2.43 and 2.44 we can define the output (or speech signal) as:

$$y_k = \sum_{i=1}^{p+q} \theta_k(i)\phi_k(i) + u_k \qquad k = 0 \ldots N-1 \qquad 3.2$$

or equivalently:

$$y_k = \phi_k^t \theta_k + u_k \qquad k = 0 \ldots N-1 \qquad 3.3$$

and:

$$\begin{aligned} \hat{y}_k &= \sum_{i=1}^{p+q} \hat{\theta}_k(i)\phi_k(i) \\ &= \phi_k^t \hat{\theta}_k \end{aligned} \qquad 3.4$$

with the estimated parameters:

$$\hat{\theta}_k = \left[\hat{a}_k(1) \quad \hat{a}_k(2) \quad \cdots \quad \hat{a}_k(p) \quad \hat{b}_k(1) \quad \hat{b}_k(2) \quad \cdots \quad \hat{b}_k(q)\right] \qquad 3.5$$

Define the cost function (or weighted recursive least squares criterion) as in 2.45, but for the prewindowed case, and introduce a weighting factor (or forgetting factor):

$$E_{N-1}(\theta) = \sum_{k=0}^{N-1} \lambda^{N-1-k} |e_k|^2 \qquad 3.6$$

An exponential weighting factor, $0 < \lambda^{N-1-k} \leq 1$, is used here as with the traditional RLS algorithm.
If $\lambda = 1$, then we have the traditional method of the least squares with infinite memory. If $\lambda < 1$, but close to one and constant, we have the traditional RLS method. The memory of the algorithm is roughly equal to $\frac{1}{1-\lambda}$ according to Haykin [56].

Stellenbosch University https://scholar.sun.ac.za

Equation 2.46, stating the principle of orthogonality, changes to:

$$\sum_{k=0}^{N-1} \lambda^{N-1-k} e_k^* \phi_k(i) = 0 \qquad i = 1 \ldots p+q \qquad 3.7$$

Note the use of the pre-windowed boundaries and the forgetting factor.
Substitute $e_k = y_k - \hat{y}_k$ with $k = 0 \ldots N-1$ and 3.4 into 3.7. The normal equation is given below:

$$\sum_{i=1}^{p+q} \hat{\theta}_k^*(i) \sum_{k=0}^{N-1} \lambda^{N-1-k} \phi_k(i) \phi_k^*(t) = \sum_{k=0}^{N-1} \lambda^{N-1-k} \phi_k(i) y_k^* \qquad 3.8$$

for $i, t = 1 \ldots p+q$

In matrix form:

$$\Phi_k \hat{\theta}_k = \Theta_k \qquad k = 0 \ldots N-1 \qquad 3.9$$

Where:

$$\Phi(i,t) = \sum_{k=0}^{N-1} \lambda^{N-1-k} \phi_k(i) \phi_k^*(t) \qquad i,t = 1 \ldots p+q \qquad 3.10$$

is the time averaged covariance matrix.
and:

$$\Theta(i) = \sum_{k=0}^{N-1} \lambda^{N-1-k} \phi_k(i) y_k^* \qquad i = 1 \ldots p+q \qquad 3.11$$

is the time averaged cross-covariance vector.

Equation 3.9 can be solved for the filter parameters by inverting the correlation matrix:

$$\hat{\theta}_k = \Phi_k^{-1} \Theta_k \qquad k = 0 \ldots N-1 \qquad 3.12$$

If the forgetting factor is kept constant and equal to unity, we have the ordinary least squares method. It is not recursive, so the covariance matrix as well as the cross-covariance vector have to be calculated from scratch at each time increment. To add to this computational expense, the covariance matrix needs to be inverted at each time step. This is a time consuming, ineffective way of solving the parameters, especially if the ARMA orders become large. Recursive formulations for updating the covariance matrix and cross-covariance vector will be presented next.

Stellenbosch University https://scholar.sun.ac.za

3.2.1. Recursion for updating the covariance matrix

The recursion for updating the covariance matrix is shown below:

$$\Phi_{N-1} = \sum_{k=0}^{N-1} \lambda^{N-1-k} \phi_k \phi_k^H \qquad 3.13$$

First, isolate the term corresponding to $k = N-1$:

$$\Phi_{N-1} = \lambda \left[\sum_{k=0}^{N-2} \lambda^{N-2-k} \phi_k \phi_k^H \right] + \phi_{N-1} \phi_{N-1}^H \qquad 3.14$$

then recognise that the term between the square brackets is the previous value of the covariance matrix:

$$\Phi_{N-1} = \lambda \Phi_{N-2} + \phi_{N-1} \phi_{N-1}^H \qquad 3.15$$

3.2.2. Recursion for updating the cross-covariance vector

It is also possible to update the cross-covariance vector in much the same way as the covariance matrix.

$$\Theta_{N-1} = \sum_{k=0}^{N-1} \lambda^{N-1-k} \phi_k y_k^* \qquad 3.16$$

Isolate the term corresponding to $k = N-1$:

$$\Theta_{N-1} = \lambda \sum_{k=0}^{N-2} \lambda^{N-2-k} \phi_k y_k^* + \phi_{N-1} y_{N-1}^* \qquad 3.17$$

Recognise that the term between the square brackets is the previous value of the cross-covariance matrix:

$$\Theta_{N-1} = \lambda \Theta_{N-2} + \phi_{N-1} y_{N-1}^* \qquad 3.18$$

3.2.3. Inverting the covariance matrix using the matrix inversion lemma

The matrix inversion lemma states that, if A and B are two positive definite $M \times M$ matrices and D is also a positive definite $N \times N$ matrix and C is a $M \times N$ matrix with the following relation between the matrices A, B, C and D:

$A = B^{-1} + CD^{-1}C^H$

then the inverse of A is given by:

$$A^{-1} = B - BC\left(D + C^H BC\right)^{-1} C^H B$$

If we now set $A = \Phi_{N-1}$, $B^{-1} = \lambda \Phi_{N-2}$, $C = \phi_{N-1}$ and $D = 1$ and use these definitions in the recursive equation for the covariance matrix then the inverse of the covariance matrix is:

$$\begin{aligned} \Phi_{N-1}^{-1} &= \lambda^{-1}\Phi_{N-2}^{-1} + \lambda^{-1}\Phi_{N-2}^{-1}\phi_{N-1}\left(1 + \phi_{N-1}^H \lambda^{-1}\Phi_{N-2}^{-1}\phi_{N-1}\right)^{-1} \phi_{N-1}^H \lambda^{-1}\Phi_{N-2}^{-1} \\ &= \lambda^{-1}\Phi_{N-2}^{-1} - \frac{\lambda^{-2}\Phi_{N-2}^{-1}\phi_{N-1}}{\left(1 + \lambda^{-1}\phi_{N-1}^H \Phi_{N-2}^{-1}\phi_{N-1}\right)} \phi_{N-1}^H \Phi_{N-2}^{-1} \end{aligned} \qquad 3.19$$

Stellenbosch University https://scholar.sun.ac.za

Define the covariance matrix as:

$$P_{N-1} = \Phi_{N-1}^{-1} \tag{3.20}$$

and the gain vector:

$$K_{N-1} = \frac{\lambda^{-1}\Phi_{N-2}^{-1}\phi_{N-1}}{\left(1+\lambda^{-1}\phi_{N-1}^{H}\Phi_{N-2}^{-1}\phi_{N-1}\right)}$$

$$K_{N-1} = \frac{P_{N-2}\phi_{N-1}}{\left(\lambda+\phi_{N-1}^{H}P_{N-2}\phi_{N-1}\right)} \tag{3.21}$$

With these definitions of the gain vector and the inverse of the covariance matrix we can rewrite equation 3.19 in a form for updating the inverse of the covariance matrix:

$$P_{N-1} = \lambda^{-1}P_{N-2} - \lambda^{-1}K_{N-1}\phi_{N-1}^{H}P_{N-2}$$

$$P_{N-1} = \lambda^{-1}\left[P_{N-2} - K_{N-1}\phi_{N-1}^{H}P_{N-2}\right] \tag{3.22}$$

From the equation for the gain vector (3.21) and equation 3.22, the definition for the gain vector is:

$$K_{N-1} + K_{N-1}\lambda^{-1}\phi_{N-1}^{H}P_{N-2}\phi_{N-1} = \lambda^{-1}P_{N-2}\phi_{N-1}$$

$$\begin{aligned} K_{N-1} &= \left(\lambda^{-1}P_{N-2} - \lambda^{-1}K_{N-1}\phi_{N-1}^{H}P_{N-2}\right)\phi_{N-1} \\ &= P_{N-1}\phi_{N-1} \end{aligned} \tag{3.23}$$

The gain vector is thus defined by the product of the tap-input vector, ϕ_k, and the inverse of the covariance matrix.

3.2.4. Updating the tap-weight vector

In this section we deduce the equation for updating the tap-weight vector. We know that:

$$\Theta_{N-1} = \lambda\Theta_{N-2} + \phi_{N-1}y_{N-1}^{*} \quad \text{and} \quad \hat{\theta}_k = \Phi_k^{-1}\Theta_k$$

thus we can write:

$$\begin{aligned} \hat{\theta}_{N-1} &= \Phi_{N-1}^{-1}\Theta_{N-1} \\ &= P_{N-1}\Theta_{N-1} \\ &= \lambda P_{N-1}\Theta_{N-2} + P_{N-1}\phi_{N-1}y_{N-1}^{*} \end{aligned}$$

Replace only the first term of P_{N-1} with its definition.

Stellenbosch University https://scholar.sun.ac.za

$$\hat{\theta}_{N-1} = \lambda\left(\lambda^{-1}\left[P_{N-2} - K_{N-1}\phi_{N-1}^{H}P_{N-2}\right]\right)\Theta_{N-2} + P_{N-1}\phi_{N-1}y_{N-1}^{*}$$
$$= \left(P_{N-2}\Theta_{N-2}\right) - K_{N-1}\phi_{N-1}^{H}\left(P_{N-2}\Theta_{N-2}\right) + P_{N-1}\phi_{N-1}y_{N-1}^{*}$$

Now replace $\hat{\theta}_{N-2} = P_{N-2}\Theta_{N-2}$ and $K_{N-1} = P_{N-1}\phi_{N-1}$.

$$\hat{\theta}_{N-1} = \hat{\theta}_{N-2} - K_{N-1}\phi_{N-1}^{H}\hat{\theta}_{N-2} + K_{N-1}y_{N-1}^{*}$$
$$= \hat{\theta}_{N-2} + K_{N-1}\left(-\phi_{N-1}^{H}\hat{\theta}_{N-2} + y_{N-1}^{*}\right)$$
$$\hat{\theta}_{N-1} = \hat{\theta}_{N-2} + K_{N-1}\left(y_{N-1}^{*} - \phi_{N-1}^{H}\hat{\theta}_{N-2}\right) \qquad 3.24$$

Define the conjugate of the innovation or *a priori* estimation error as the term between brackets :

$$\alpha_{N-1}^{*} = y_{N-1}^{*} - \phi_{N-1}^{H}\hat{\theta}_{N-2}$$
$$\alpha_{N-1} = y_{N-1} - \phi_{N-1}^{t}\hat{\theta}_{N-2}^{*} \qquad 3.25$$

The *a posteriori* estimation error is then defined as :

$$e_{N-1} = y_{N-1} - \phi_{N-1}^{t}\hat{\theta}_{N-1}^{*} \qquad 3.26$$

Recognise this as the same error that we used originally in the cost function.

3.2.5. Weighted RLS algorithm (with constant forgetting factor)

The WRLS algorithm can be summarised as follows:

Innovation:	$\alpha_k = y_k - \phi_k^t \hat{\theta}_{k-1}^*$
Update gain vector:	$K_k = \frac{P_{k-1}\phi_k}{\left(\lambda + \phi_k^H P_{k-1}\phi_k\right)}$
Update parameters:	$\hat{\theta}_k = \hat{\theta}_{k-1} + K_k\left(y_k^* - \phi_k^H \hat{\theta}_{k-1}\right)$
Update covariance matrix:	$P_k = \lambda^{-1}\left[P_{k-1} - K_k \phi_k^H P_{k-1}\right]$
Input estimate:	$\hat{u}_k = y_k - \phi_k^t \hat{\theta}_k^*$

3.3. VARIABLE FORGETTING FACTOR (VFF) ALGORITHM

During the derivation of the weighted RLS with constant exponential weighting factor we assumed that $0 < \lambda \leq 1$ and $\lambda = \lambda_k = \lambda_{k-1}$ for $k = 0 \ldots N-1$.

We shall now derive an algorithm for varying the length of the memory of the weighted RLS algorithm. A VFF will enable us to select a forgetting factor close to unity for stationary signals or a smaller forgetting factor for non-stationary signals. The first step will be to obtain a recursive equation for the cost function or error information based on the estimation error (*a posteriori* error).

$$E_{N-1}(\theta) = \sum_{k=0}^{N-1} \lambda^{N-1-k} |e_k|^2$$

Isolate the term corresponding to $N-1$.

$$E_{N-1}(\theta) = \lambda \sum_{k=0}^{N-2} \lambda^{N-2-k} |e_k|^2 + |e_{N-1}|^2$$

Recognise that the term in the summation represents the previous value of the cost function.

$$E_{N-1}(\theta) = \lambda E_{N-2}(\theta) + |e_{N-1}|^2 \qquad 3.27$$

Write the estimation error in terms of the gain vector and the innovation.

Stellenbosch University https://scholar.sun.ac.za

$$
\begin{aligned}
e_k &= y_k - \phi_k^t \hat{\theta}_k \\
&= y_k - \phi_k^t \left(\hat{\theta}_{k-1} + K_k \alpha_k \right) \\
&= y_k - \phi_k^t \hat{\theta}_{k-1} - \phi_k^t K_k \alpha_k \\
&= \alpha_k - \phi_k^t K_k \alpha_k \\
e_k &= \left(1 - \phi_k^t K_k\right) \alpha_k \qquad 3.28
\end{aligned}
$$

Equation 3.27 now becomes:

$$E_k(\theta) = \lambda E_{k-1}(\theta) + \left(1 - \phi_k^t K_k\right)^2 \alpha_k^2$$

But α_k^2 is a scalar, thus:

$$E_k(\theta) = \lambda E_{k-1}(\theta) + \alpha_k^2 \left(1 - \phi_k^t K_k\right)^2 \qquad 3.29$$

Define the variable forgetting factor, λ_k, so that it will compensate for the new error information at each step k. Thus we have $E_k = E_{k-1} = \ldots = E_0$. Set $\lambda = \lambda_k$ in 3.29 and isolate λ_k on the left hand side:

$$\lambda_k = \left[E_k(\theta) - \alpha_k^2 \left(1 - \phi_k^t K_k\right)^2 \right] \Big/ E_{k-1}(\theta)$$

Remember that $E_k = E_{k-1} = \ldots = E_0$ so that the VFF (variable forgetting factor) is given by:

$$\lambda_k = 1 - \alpha_k^2 \left(1 - \phi_k^t K_k\right)^2 / E_0(\theta) \qquad 3.30$$

Notice that, when the estimation error (e_k) is small, the value of λ_k will be close to unity. This implies a long memory, because Haykin [56] defines the memory of the algorithm as $M = \dfrac{1}{1-\lambda}$. This is exactly what we want when analysing stationary signals. For a large estimation error the value of λ_k will be smaller than unity, implying a shorter memory length. This will allow faster tracking in non-stationary environments.

Note however that for some applications we may need a certain minimum memory size. In these cases it is recommended that a minimal memory length be defined as being equal to twice the sum of the poles and the zeros [56] of the AR and MA models respectively.

$$\lambda_{min} = \frac{2(p+q) - 1}{2(p+q)}$$

where p=poles in AR model and q=zeros in MA model.

Stellenbosch University https://scholar.sun.ac.za

Ways to select an appropriate value for E_0 (the error information) will be discussed in chapter 4.

3.4. WHITE NOISE & PULSE INPUT ALGORITHM

When deriving the classical RLS algorithm it was assumed that the input signal (u_k) to the filter is a zero mean, white, gaussian noise process. This is however not true when modelling the speech process. Two input models are commonly used for modelling speech. A pulse input signal is generally assumed for vowel sounds and a white noise input signal for generating fricative sounds. We shall now use the symbol u_k^w to represent a white noise input signal and u_k^p for a pulse input sequence. Thus the total resulting input signal is:

$$u_k = u_k^w + u_k^p \qquad 3.31$$

With the estimation error defined as follows:

$$e_k = y_k - \hat{y}_k - \hat{u}_k$$

and cost function:

$$E_{N-1}(\theta) = \sum_{k=0}^{N-1} \lambda^{N-1-k} |e_k|^2$$

If we assume ergodicity then $\sum_{k=0}^{N-1} u_k^w = \mathcal{E}\{u_k^w\}$ if $N \to \infty$. This implies that the cost function will only contain the pulse-related part of the input sequence, because

$$\sum_{k=0}^{N-1} u_k = \sum_{k=0}^{N-1} u_k^p \qquad \text{if } N \to \infty$$

The following cost function has to be minimized:

$$E_{N-1}(\theta) = \sum_{k=0}^{N-1} \lambda^{N-1-k} \left| y_k - \hat{y}_k - \hat{u}_k^p \right|^2 \qquad 3.32$$

Substitute $e_k = y_k - \hat{y}_k - \hat{u}_k^p$ with $k = 0 \ldots N-1$ and 3.4 into 3.32. The normal equation is given below:

$$\sum_{i=1}^{p+q} \hat{\theta}_k^*(i) \sum_{k=0}^{N-1} \lambda^{N-1-k} \phi_k(i) \phi_k^*(t) + \sum_{k=0}^{N-1} \lambda^{N-1-k} \phi_k(i) \hat{u}_k^{*p} = \sum_{k=0}^{N-1} \lambda^{N-1-k} \phi_k(i) y_k^* \qquad 3.33$$

for $i, t = 1 \ldots p+q$

In matrix form:

Stellenbosch University https://scholar.sun.ac.za

$$\Phi_k \hat{\theta}_k + \Psi_k = \Theta_k \qquad k = 0 \ldots N-1 \qquad 3.34$$

Where:

$$\Psi(i) = \sum_{k=0}^{N-1} \lambda^{N-1-k} \phi_k(i) \hat{u}_k^{*p} \qquad i = 1 \ldots p+q \qquad 3.35$$

is the time averaged cross-covariance vector due to a pulse input signal.

The equations for updating the covariance matrix and gain vector stay the same as in the case of the WRLS-VFF without the input estimation scheme. The derivation for updating the parameters will follow:

$$\begin{aligned} \hat{\theta}_{N-1} &= \Phi_{N-1}^{-1}\Theta_{N-1} - \Phi_{N-1}^{-1}\Psi_{N-1} \\ &= P_{N-1}\Theta_{N-1} - P_{N-1}\Psi_{N-1} \\ &= P_{N-1}\left[\Theta_{N-1} - \Psi_{N-1}\right] \end{aligned} \qquad 3.36$$

Replace $\Theta_{N-1} = \lambda\Theta_{N-2} + \phi_{N-1}y_{N-1}^*$ and $\Psi_{N-1} = \lambda\Psi_{N-2} + \phi_{N-1}\hat{u}_{N-1}^{*p}$ to obtain:

$$\begin{aligned} \hat{\theta}_{N-1} &= P_{N-1}\left[\lambda\Theta_{N-2} + \phi_{N-1}y_{N-1}^* - \lambda\Psi_{N-2} - \phi_{N-1}\hat{u}_{N-1}^{*p}\right] \\ &= \lambda P_{N-1}\Theta_{N-2} + P_{N-1}\phi_{N-1}y_{N-1}^* - \lambda P_{N-1}\Psi_{N-2} - P_{N-1}\phi_{N-1}\hat{u}_{N-1}^{*p} \end{aligned}$$

Replace $P_{N-1} = \lambda^{-1}\left[P_{N-2} - K_{N-1}\phi_{N-1}^H P_{N-2}\right]$ in the terms containing the forgetting factor.

$$\begin{aligned} \hat{\theta}_{N-1} &= P_{N-2}\Theta_{N-2} - K_{N-1}\phi_{N-1}^H P_{N-2}\Theta_{N-2} + P_{N-1}\phi_{N-1}y_{N-1}^* \\ &\quad - P_{N-2}\Psi_{N-2} - K_{N-1}\phi_{N-1}^H P_{N-2}\Psi_{N-2} - P_{N-1}\phi_{N-1}\hat{u}_{N-1}^{*p} \end{aligned}$$

Replace $\hat{\theta}_{N-2} = P_{N-2}\left[\Theta_{N-2} - \Psi_{N-2}\right]$ (from equation 3.36).

$$\hat{\theta}_{N-1} = \hat{\theta}_{N-2} - K_{N-1}\phi_{N-1}^H\hat{\theta}_{N-2} + P_{N-1}\phi_{N-1}y_{N-1}^* - P_{N-1}\phi_{N-1}\hat{u}_{N-1}^{*p}$$

We also recognize a further simplification by using $K_{N-1} = P_{N-1}\phi_{N-1}$.

$$\hat{\theta}_{N-1} = \hat{\theta}_{N-2} - K_{N-1}\phi_{N-1}^H\hat{\theta}_{N-2} + K_{N-1}y_{N-1}^* - K_{N-1}\hat{u}_{N-1}^{*p}$$

Isolate K_{N-1} from the last three terms.

$$\hat{\theta}_{N-1} = \hat{\theta}_{N-2} + K_{N-1}\left(y_{N-1}^* - \phi_{N-1}^H\hat{\theta}_{N-2} - \hat{u}_{N-1}^{*p}\right) \qquad 3.37$$

We now have the equation for updating the parameters when the input to the filter is defined as either white noise or periodic pulses. Note that by subtracting the pulse input signal from the new estimation for the parameters, the influence of pitch pulses can be removed.

Stellenbosch University https://scholar.sun.ac.za

In their work of 1982, Morikawa and Fujisaki [7] use the estimated error, $e_k = y_k - \phi_k^t \hat{\theta}_k$, as an input signal to their SEARMA method (Simultaneous Estimation ARMA). They assumed, like we did when developing the WRLS, that the excitation source of the vocal tract produces purely white noise.

Miyanaga *et al.* [11] proposed a method to estimate both white noise and pulses for the input signal. They used two adaptive algorithms to do this. The first algorithm is used to estimate the parameters and compute the optimal estimation error and its covariance. The second algorithm is used to compute the actual estimation error and, by using the covariance of the optimal estimation error, decides whether the input is white noise or a pulse. In a later paper [12] they extended their method to include estimation of non-stationary parameters.

We shall use the method proposed by Ting and Childers [14]. It is similar to the one of Miyanaga *et al.* [11], but uses the VFF to decide on white noise or pulse input. Thus only one instead of two adaptive processes is required.

During the derivation of the VFF we pointed out that, for a large estimation error, the VFF becomes small and vice versa. A sudden very small value for the VFF will indicate the occurrence of a pitch pulse. By defining a lower threshold for the forgetting factor (λ_0) we can select a pulse input signal if $\lambda_k < \lambda_0$ and a white noise input for $\lambda_k \geq \lambda_0$.

Miyanaga *et al.* [11] showed that the magnitude of the pulse is approximately the same as that of the innovation. Thus we have $\hat{u}_k^p = y_k - \phi_k^t \hat{\theta}_{k-1}$ and $\hat{u}_k^w = 0$ if $\lambda_k < \lambda_0$.

The white noise input is selected when $\lambda_k \geq \lambda_0$ by using the method of Morikawa and Fujisaki [7]. Thus we have $\hat{u}_k^p = 0$ and $\hat{u}_k^w = y_k - \phi_k^t \hat{\theta}_k = \alpha_k \left(1 - \phi_k^t K_k\right)$.

Stellenbosch University https://scholar.sun.ac.za

3.4.1. WRLS-VFF algorithm with input estimation

The WRLS-VFF algorithm with input estimation is summarised as follows:

Innovation: $\alpha_k = y_k - \phi_k^t \hat{\theta}_{k-1}^*$

Update gain vector: $K_k = \dfrac{P_{k-1}\phi_k}{\left(\lambda + \phi_k^H P_{k-1}\phi_k\right)}$

Update forgetting factor: $\lambda_k = 1 - \alpha_k^2\left(1 - \phi_k^t K_k\right)^2 / E_0(\theta)$

Input estimation: If $\lambda_k < \lambda_0$ (Pulse input)

$$u_k^w = 0$$
$$u_k^p = y_k - \phi_k^t \hat{\theta}_{k-1}^*$$

If $\lambda_k \geq \lambda_0$ (White noise input)

$$u_k^w = \alpha_k\left(1 - \phi_k^t K_k\right)$$
$$u_k^p = 0$$

Update parameters: $\hat{\theta}_k = \hat{\theta}_{k-1} + K_k\left(y_k^* - \phi_k^H \hat{\theta}_{k-1} - \hat{u}_k^{*p}\right)$

Update covariance matrix: $P_k = \lambda^{-1}\left[P_{k-1} - K_k \phi_k^H P_{k-1}\right]$

The block diagram of the WRLS-VFF is shown in figure 3.8.

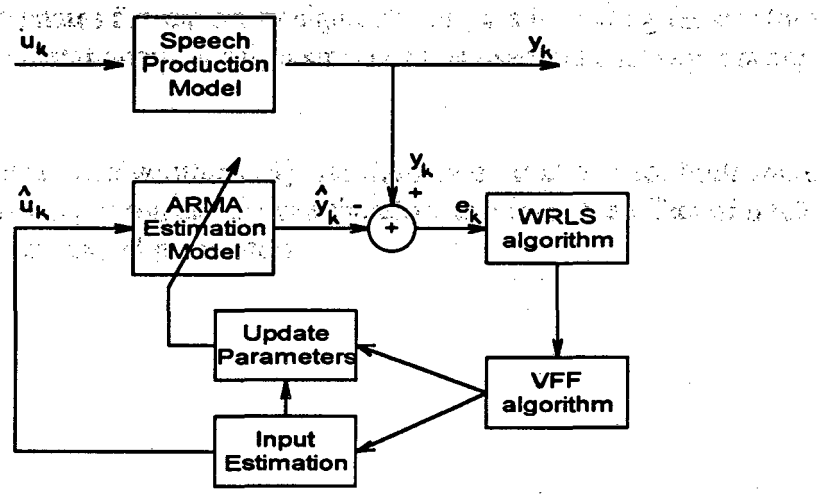


Figure 3.8 Blockdiagram of the WRLS-VFF algorithm.

3.5. COMPUTATIONAL REQUIREMENTS

The complexity analysis of the WRLS-VFF is presented in Appendix D. The algorithm requires approximately $5(p+q)^2+9(p+q)$ multiplications and additions (flops) per data point. If we compare this with the complexity of the Marple block technique (of pN flops per block), it is clear that with increasing data samples (N) the computation for the WRLS-VFF will become excessively large. This is the main disadvantage of the WRLS-VFF. However, if a fast algorithm is developed for the WRLS-VFF, it may reduce the computational complexity to O(pN) for analysing N data points. The least squares lattice (LSL) algorithm by Lee *et al* [34] already does this for the normal weighted RLS method. Such a fast algorithm will have to provide a more efficient way of computing the update of the covariance matrix, which takes up the largest slice of the computational time. In a method proposed by Gavrishchuk and Starkov [33] they claim to do exactly this, namely to reduce the number of operations required for calculating the covariance matrix of , the estimation errors.

3.6. CONCLUSION

The contributions of this chapter are:

- A recursive ARMA parameter estimation algorithm, the WRLS-VFF [14], was derived.
- The built-in VFF enables the algorithm to obtain asymptotically unbiased parameter estimates for stationary signals by using a forgetting factor close to unity. For non-stationary signals a smaller FF is used, thus allowing faster tracking.
- An input estimation algorithm [14] was described. It allows for both white noise and pulse input signals and can therefore eliminate the effect of pitch pulses on the estimated parameters.

Stellenbosch University https://scholar.sun.ac.za

Chapter 4

Implementing the WRLS-VFF

4.1. INTRODUCTION

In this chapter some practical procedures for the implementation of the WRLS-VFF algorithm to estimate ARMA speech parameters will be discussed. Techniques for choosing the order of the ARMA process are suggested. Initiation of the covariance matrix and the initial parameter values is discussed. A simple, yet effective, empirical algorithm is introduced to automatically determine the value of the error information. Minimum and threshold values for the forgetting factor are motivated. At the end of the chapter we include possible techniques to improve the tracking of certain nonstationary signals.

4.2. MODEL ORDER SELECTION

While deriving the WRLS-VFF in chapter 3 it was assumed that the order of the unknown ARMA process was known beforehand. In this section we shall explore some practical ways for determining the order of the ARMA process before the WRLS-VFF is started.

In parametric estimation, choosing the correct underlying model order has important implications regarding the accuracy of the estimated parameters ([10], [14], [51] and [50]). It is known that, if the AR order selected is too low, there will be too few poles in the estimated spectrum to adequately represent the real speech model. On the other hand, too many poles in the estimated model will result in spurious peaks in the spectrum. A wrong estimate of the MA order may influence the estimates of the poles as well as that of the zeros ([51], [50]).

Various order determination techniques have been proposed over the years. Akaike introduced the final prediction error criterion (FPE) and later extended it to the Akaike information criterion [28]. Rissanen [27] proposed the minimum description length (MDL) criterion, which is an extension of the AIC criterion. Parzen [29] introduced the criterion autoregressive transfer function (CAT).

Pukkila and Krishnaiah [32] showed that most of these criteria can be expressed in

Stellenbosch University https://scholar.sun.ac.za

the form:

$$Method_{p,q} = N\log(\sigma_{\hat{u}}^2) + (p+q)\beta$$

where p and q are the orders of the AR and MA processes respectively. $\sigma_{\hat{u}}^2$ is the variance of the estimation error and β is a nonnegative penalty term. The optimal order is found by minimizing this function.

The AIC criterion is then defined by replacing $\beta = 2$:

$$AIC_{p,q} = N\log(\sigma_{\hat{u}}^2) + 2(p+q)$$

The MDL criterion is defined by using $\beta = \log(N)$

$$MDL_{p,q} = N\log(\sigma_{\hat{u}}^2) + \log(N)(p+q)$$

Hannan [31] proved that the AIC criterion produces inconsistent estimates for the ARMA model with white noise excitation. He also showed that the MDL criterion does produce consistent estimates of this model. The MDL criterion will thus be used in the discussion that follows.

Morikawa [10], Miyanaga [12], and Ting and Childers [14] proposed methods for determining the orders of an ARMA process. The author suggests a combination of these:

- Determine the AR order by applying the MDL criterion to the first N samples of the speech.
- Use this constant AR order in the ARMA WRLS-VFF algorithm while increasing the MA order from zero.
- Obtain the smoothed spectrum by using the cepstral method described in chapter 6.
- Find the spectral distance between the smoothed spectrum and the ARMA spectrum.
- The optimum value for the MA part is selected when the spectral distance is a minimum.

This procedure for estimating the ARMA order of the model is not optimal, because we determine the AR and MA orders separately. The correct procedure would be to determine both the AR and MA orders simultaneously. However, Morikawa and Fujisaki showed that an order estimation error of ±2 does not generate large discrepancies in the estimation of the pole and zero frequencies. If the above procedure is selected over the optimal method, a maximum of p+q passes of the WRLS-VFF algorithm are needed instead of the otherwise $p \times q$ passes.

4.3. INITIALIZATION

A few variables need to be set at initiation of the WRLS-VFF:

- According to Ting and Childers [14] and Haykin [56] the initial value of the covariance matrix may be set to $P_0 = \sigma \mathbf{I}$, with σ a large positive number. A too small value for σ will slow down the rate of convergence of the algorithm. Morikawa and Fujisaki [7] showed however that the convergence properties are not significantly affected, as long as P_0 is large compared to the variance of the source signal, ϕ_t. These results were verified by a statistical analysis on the initiation of the RLS algorithm by Hubing and Alexander [16]. In summary, we can say that the choice of P_0 is based on practical experience with the RLS algorithm. For large data records the exact value is, however, unimportant.

- The initial estimation of the parameter vector may be set to zero ($\hat{\theta}_0 = 0$).

- The error information, E_0 (sum of the estimation errors), can be calculated before the WRLS-VFF algorithm is started. Ting and Childers [14] suggested use of a LPC method on a frame or two to determine a suitable value. In their tests on short segments of speech they used a constant value of 10^6 for estimating the speech-parameters. They also stated that the algorithm is insensitive to the value E_0, as long as it is large enough. *The author discovered that this is not true and shall explain the reasons for this in the next section.*

- A minimum value for the forgetting factor can be defined. This will prevent the memory of the algorithm from becoming shorter than a specified number of samples. We follow the recommendation of Haykin [56] and choose:

$$\lambda_{\min} = \frac{2(p+q)-1}{2(p+q)}$$

If $\lambda < \lambda_{\min}$ then we set $\lambda = \lambda_{\min}$. This value of $\lambda_{\min}$ corresponds to a memory length of $2(p+q)$ samples, which is the minimum that is required for convergence of the RLS algorithm [56].

- The threshold value for detecting different input signals was determined experimentally. We use a value of $\lambda_0 = \lambda_{\min} + .01$ throughout the rest of this document.

Stellenbosch University https://scholar.sun.ac.za

4.4. COMPUTING THE FORGETTING FACTOR

Ting and Childers [14] tested the WRLS-VFF only on short segments of speech. In their paper they concluded that the algorithm is insensitive to the error information, E_0, as long as its value is large enough. If we look at the equation for the variable forgetting factor (3.30), i.e.: $\lambda_k = 1 - \alpha_k^2 \left(1 - \phi_k' K_k\right)^2 / E_0(\theta)$ we see that if E_0 becomes excessively large ($E_0 >> |e_k|^2$) then the memory of the algorithm is close to infinite ($\lambda_k \approx 1$). In such a case the parameter values will be identical to that obtained by the ordinary least squares method, without a forgetting factor! This is clearly not what we intend with the WRLS-VFF.

The memory of the WRLS-VFF algorithm is influenced by two factors (Hubing [15]):

(1) The value of the error information which is set at constant at the initiation of the algorithm.
(2) The detected degree of stationarity of the parameters being tracked.

If the algorithm is tested on short speech utterances, the appropriate constant value for the error information can be obtained by using an LPC technique to determine the sum of the estimated errors over one or two frames. When extracting parameters of a large speech database, this is not a practical option. The author therefore considered various ways of automatically setting the value of the error information. One of these will be discussed next.

After testing the WRLS-VFF on speech by using a constant value for the error information, we noticed that tracking deteriorated in the unvoiced regions. The problem was that the specific constant value of E_0 was too high for tracking the low energy signals. Although the choice of E_0 could be perfect for the voiced (and thus higher energy) speech, it caused λ_k to be very close to unity when the estimated error (e_k) becomes small. The reader might argue that this is exactly what we want - a longer memory during times where the estimation error is small - and we agree! The magnitude of the estimation error is however related to that of the speech signal being followed. Thus, for a softly pronounced part of the speech the estimation error would be less than would be the case if the same segment is spoken in a louder voice. The above fact lead me to the following equation to determine the value of E_0 at each instant k.

$$E_k = |e_k|^2 + 7 \times G_k$$

Where G_k is the gain of the estimated ARMA filter and corresponds to the standard deviation of the estimated white noise input signal. The recursive computation of

Stellenbosch University https://scholar.sun.ac.za

the G_k will be discussed in a later paragraph. The constant multiplier of seven (in front of G_k) was determined experimentally.

4.5. COMPUTING THE PARAMETERS

Initially the algorithm skips p (the number of poles) samples at the beginning of the analysis interval. This is done in order to fill the elements of the input vector so that:

$$\begin{aligned}\phi_k' &= \begin{bmatrix} -y_{p-1} & -y_{p-2} & \cdots & -y_0 & u_{p-1} & u_{p-2} & \cdots & u_{p-q} \end{bmatrix} \\ &= \begin{bmatrix} -y_{p-1} & -y_{p-2} & \cdots & -y_0 & 0 & 0 & \cdots & 0 \end{bmatrix}\end{aligned}$$

The computation of the ARMA parameters thus start at time k=p.

4.6. COMPUTING THE GAIN OF THE ARMA FILTER

Take another look at the ARMA equation 2.3 with $\hat{b}_k(0) = 1$ and $\hat{a}_k(0) = 1$:

$$y_k = -\sum_{i=1}^{p} \hat{a}_k(i) y_{k-i} + \sum_{i=1}^{q} \hat{b}_k(i) \hat{u}_{k-i} + \hat{u}_k$$

Here the filter input $\hat{u}_k$ is scaled to account for any filter gain. In such a case the sequence $\hat{u}_k$ will not have unit variance anymore, but a variance of $\sigma_{\hat{u}_k}^2$. Let us rewrite 2.3 in a form where $\hat{u}_k$ has a unit variance, but without changing the value of $\hat{b}_k(0) = 1$. We do this by normalising $\hat{u}_k$ with its standard deviation $\sigma_{\hat{u}}$, at time k, and introducing a gain factor G_k:

$$y_k = -\sum_{i=1}^{p} \hat{a}_k(i) y_{k-i} + \sum_{i=1}^{q} \hat{b}_k(i) G_k \hat{u}_{k-i}' + G_k \hat{u}_k'$$

Note that the gain sequence, G_k, can change with time, as the variance of $\hat{u}_k$ changes. Let $\hat{u}_k = G_k \hat{u}_k'$, then the variance of $\hat{u}_k$ is $\sigma_{\hat{u}_k}^2 = G_k^2 \sigma_{\hat{u}_k'}^2 = G_k^2$. The value of G_k is the standard deviation of the estimated input to the filter and is also the gain of the estimated ARMA filter at time k.

Stellenbosch University https://scholar.sun.ac.za

Recursive computation of the gain of the filter

For an infinite memory length, the gain of the filter can be computed recursively using the following equation (Oppenheim and Schafer [55]):

$$\sigma_{\hat{u}_k}^2 = \frac{N}{N-1}\left[\frac{1}{N}\sum_{k=1}^{N}\hat{u}_k^2 - \left(\frac{1}{N}\sum_{k=1}^{N}\hat{u}_k\right)^2\right]$$

$$\sigma_{\hat{u}_k}^2 = \frac{1}{N-1}\sum_{k=1}^{N}\hat{u}_k^2 - \frac{1}{N(N-1)}\left(\sum_{k=1}^{N}\hat{u}_k\right)^2 \qquad N>1 \qquad 4.1$$

The author modified this equation for computing the gain of the filter recursively by using a sliding window of constant length (i.e. shorter memory). For $k \le Window$ we use the following definitions:

$$s_N = \sum_{k=1}^{N}\hat{u}_k^2 = \sum_{k=1}^{N-1}\hat{u}_k^2 + \hat{u}_N^2 \quad \text{or} \quad s_k = s_{k-1} + \hat{u}_k^2 \quad \text{for } k \le Window$$

$$m_N = \sum_{k=1}^{N}\hat{u}_k = \sum_{k=1}^{N-1}\hat{u}_k + \hat{u}_N \quad \text{or} \quad m_k = m_{k-1} + \hat{u}_k \quad \text{for } k \le Window$$

For $k > Window$ we use:

$$s_k = s_{k-1} + \hat{u}_k^2 - \hat{u}_{k-Window}^2 \quad \text{and} \quad m_k = m_{k-1} + \hat{u}_k - \hat{u}_{k-Window}$$

The variance of $\hat{u}_k$ (and thus its standard deviation) can then be obtained by evaluating the following recursive equation for k = 1 ... N:

$$\sigma_{\hat{u}_k}^2 = \begin{cases} 0 & k=1 \\ \dfrac{s_k}{Window-1} - \dfrac{m_k^2}{Window\,(Window-1)} & k > \text{Window} \\ \dfrac{s_k}{k-1} - \dfrac{m_k^2}{k(k-1)} & k \le \text{Window} \end{cases}$$

Where Window=the length (in samples) of the sliding window used in the gain computation. The choice of the window length is important. If the window is too long, the gain of the filter will vary slowly with time, and fluctuations over voiced/unvoiced regions may be missed. On the other hand, if the window is too short, peaks might occur in the gain sequence as a result of pitch pulses in the voiced regions. In our tests on real speech we chose the window length to be at least two pitch periods in length.

Stellenbosch University https://scholar.sun.ac.za

4.7. IMPROVING PARAMETER TRACKING

In applying a linear technique, like the RLS, there is always the possibility of a rapid change in the speech signal to be followed. In these circumstances a linear approach is incapable of tracking the signal across the discontinuity. Typical manifestations of such discontinuities in speech occur when voiced speech is followed by an unvoiced region (i.e. fricative sounds like /f/, /s/, /w/, /v/). Although a linear method cannot follow the signal when the discontinuity occurs, it can follow a non-stationary signal afterwards, if the adaptation method is fast enough.

Morikawa and Fujisaki [7] showed that the time for convergence of the RLS algorithm in regions of unvoiced speech is much longer than in voiced speech. They specify a convergence interval of about 5 times the ARMA order for voiced speech and about 20 times the ARMA order for unvoiced speech. Although the WRLS-VFF algorithm converges much quicker than the RLS method used by Morikawa and Fujisaki, it still converges more slowly in unvoiced regions than in voiced parts. This is true even if the error information is chosen with

$E_k = |e_k|^2 + 7 \times G_k$.

If we can determine the onset of an unvoiced speech region accurately and then reset the covariance matrix to its initial value (at this point) the memory of the algorithm will be cleared. If σ (in $P_0 = \sigma \mathbf{I}$) is chosen large enough, then convergence will be much faster than can be achieved by just varying the forgetting factor if the memory is retained.

We shall now discuss a very effective method for determining voiced/unvoiced boundaries in speech.

A Fractal Method

In applying the WRLS-VFF to continuous speech, it became necessary to develop a way to detect voiced/unvoiced jumps in the speech signal. A discontinuity from a voiced to unvoiced part of speech can be defined as a place in the speech signal where the WRLS-VFF will loose track of the signal.

One way to detect these instances is to compare the estimated signal to the original speech. When the error between these two signals becomes larger than a predefined limit, a discontinuity, that cannot be followed fast enough, is defined. Another method will be to count the zero crossings in the original speech signal. The start of a region where the count is high can then be defined as a voiced/unvoiced boundary.

We propose a method based on work done by Boshoff [44] and Boshoff & Lehmensiek [46]. The idea is to determine the local fractal dimension of the sampled speech signal by using a fast box count algorithm. A value for the fractal

Stellenbosch University https://scholar.sun.ac.za

dimension greater than a predefined threshold indicates a discontinuity. The complexity of the box count algorithm compares favourably with that of zero crossing rate [44]. A further advantage is that the mean of the signal is not needed in the computation as in the case of zero crossing rate. It is also much easier to determine a threshold value for the fractal dimension, than a threshold value for the error between the estimated and original signals.

Figure 4.9 and 4.11 show part of a speech sentence ("Even my sense of humour could evolve a better"). The corresponding fractal dimension, as estimated by using the above fast box count technique, is shown in figures 4.10 and 4.12.

Stellenbosch University https://scholar.sun.ac.za

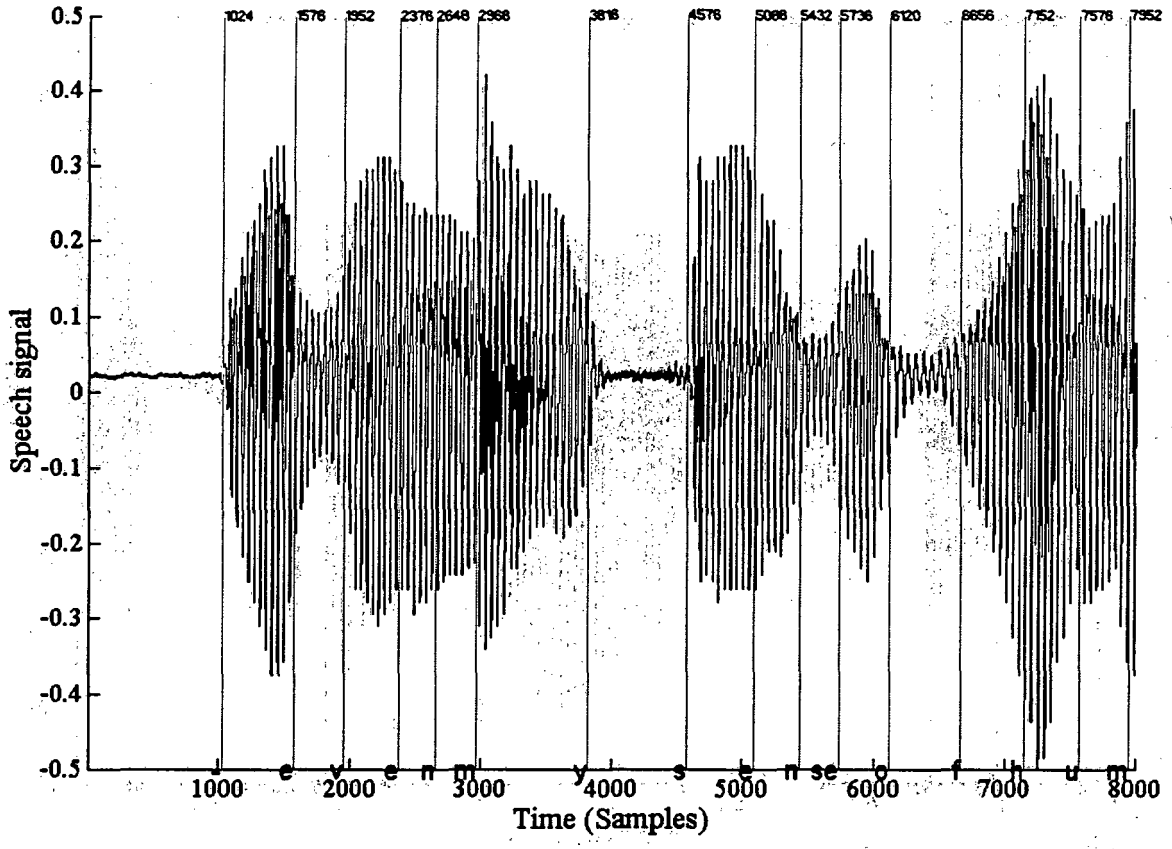


Figure 4.9 Speech signal of "even my sense of hum-".

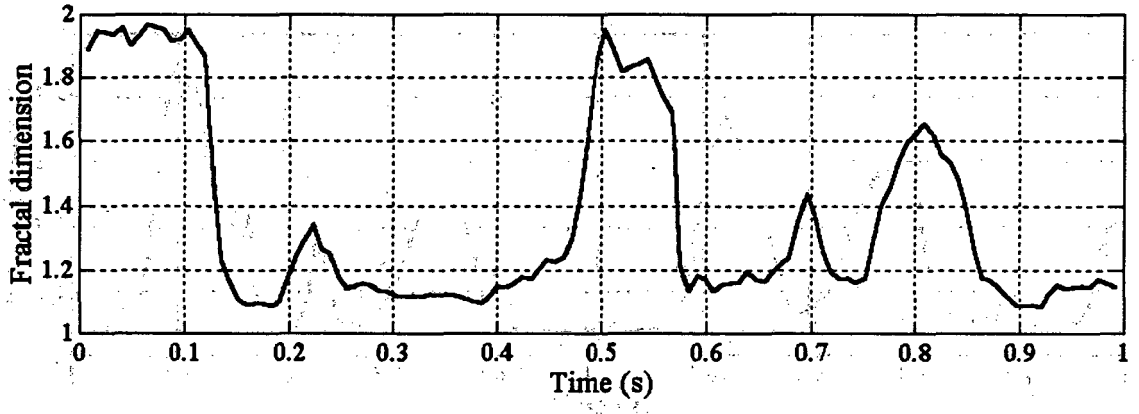


Figure 4.10 Fractal dimension of "even my sense of hum-".

Stellenbosch University https://scholar.sun.ac.za

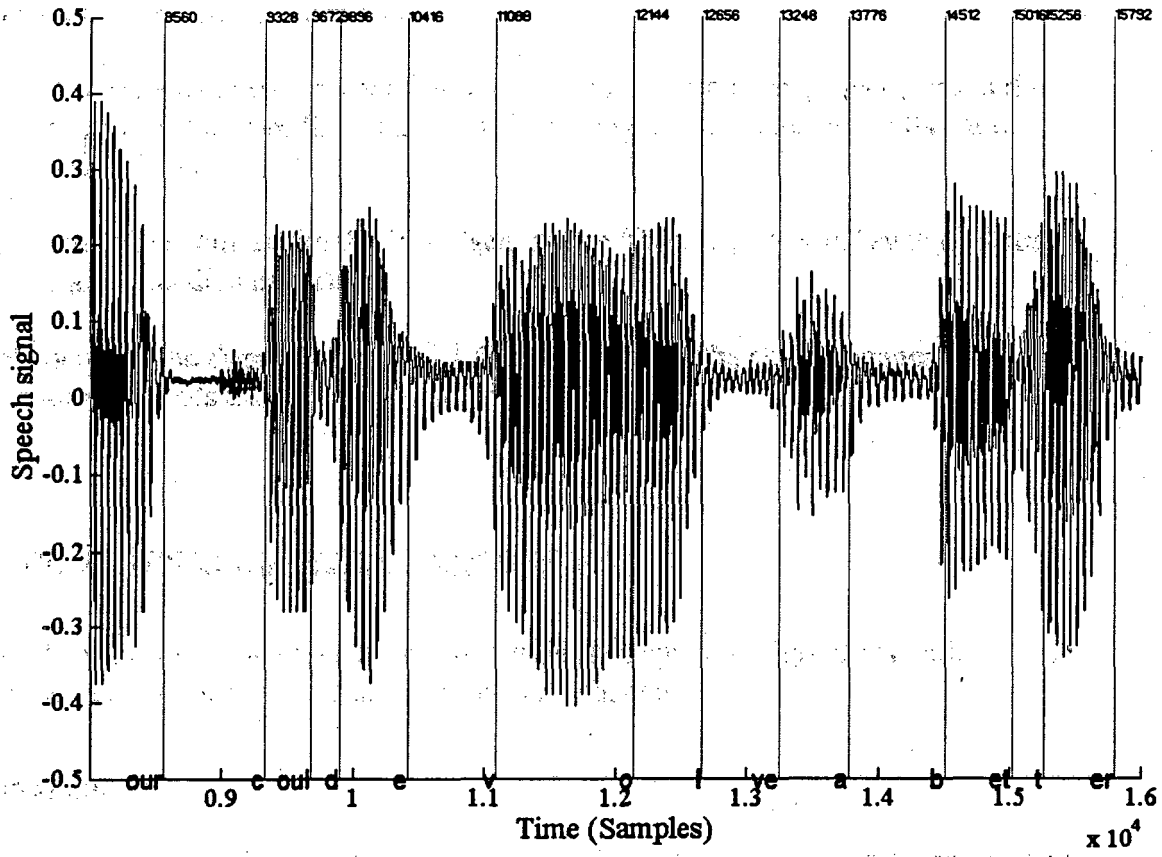


Figure 4.11 Speech signal of "-our could evolve a better".

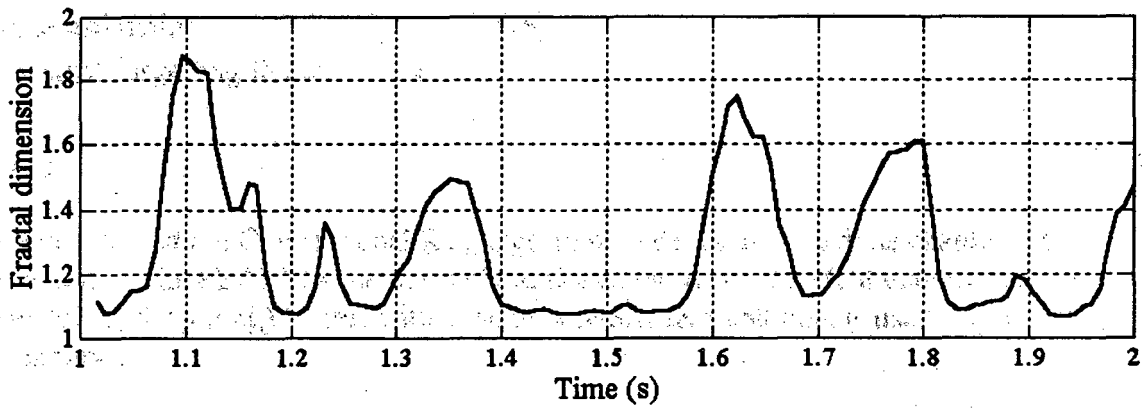


Figure 4.12 Fractal dimension of "-our could evolve a better".

Stellenbosch University https://scholar.sun.ac.za

From these results it is clear that:

- The fractal dimension becomes a value close to two during silent parts in the speech signal. See for instance the beginning of the sentence and the stop sound /c/ in "could".

- Unvoiced sounds like the /s/ in "sense", the /f/ on "of", /ve/ in "evolve" force the fractal dimension up.

By defining the threshold value as 1.6, we see that all the above mentioned unvoiced sounds and silences will be discovered.

4.8. COMPARING DIFFERENT ALGORITHMS

In the next few paragraphs we discuss different sequential algorithms with the aim of comparing their performances in the next chapter.

SEARMA

The Simultaneous Estimation of AutoRegressive and Moving-Average parameters (SEARMA-method) was first introduced by Morikawa and Fujisaki [7] in 1982. This algorithm is basically the same as the conventional RLS algorithm, but with the forgetting factor a constant and equal to unity. The input estimation algorithm assumes only a white noise input. Thus to summarise:

Input estimation: $\hat{u}_k = y_k - \phi_k^t \hat{\theta}_k^*$

Constant forgetting factor: $\lambda = 1$

CRLS

We considered the Conventional RLS algorithm in detail in a previous chapter. It differs from the SEARMA method only to the extent that a constant exponential weighting (forgetting) factor, with a value between zero and one, is used. In summary:

Input estimation: $\hat{u}_k = y_k - \phi_k^t \hat{\theta}_k^*$

Constant forgetting factor: $0 < \lambda < 1$

MRLS

The Modified WRLS algorithm is an CRLS algorithm, but with a variable forgetting factor [14]. The error information, E_0, is chosen at initiation and stays constant throughout the adaptation process (see the derivation of the variable forgetting factor). The input process is still a white noise process. We summarise the MRLS method:

Stellenbosch University https://scholar.sun.ac.za

Input estimation: $\hat{u}_k = y_k - \phi_k^t \hat{\theta}_k$
Variable forgetting factor: $0 < \lambda < 1$
Error information: constant, chosen at initiation.

WRLS-VFF

The Weighted RLS algorithm with a Variable Forgetting Factor was originally developed by Ting and Childers [14]. The detail of the algorithm is covered extensively throughout this document. It differs from the MRLS method in that the input estimation algorithm allows for a white noise or a pulse input signal. The summary follows:

Input estimation: If $\lambda_k < \lambda_0$ (Pulse input)

$$u_k^w = 0$$
$$u_k^p = y_k - \phi_k^t \hat{\theta}_{k-1}$$

If $\lambda_k \geq \lambda_0$ (White noise input)

$$u_k^w = \alpha_k \left(1 - \phi_k^t K_k\right)$$
$$u_k^p = 0$$

Variable forgetting factor: $0 < \lambda_k < 1$
Error information: constant, chosen at initiation.

FWRLS

The Fractal WRLS is a WRLS-VFF algorithm, with a fast box count algorithm (Boshoff [44]) included to allow better tracking of the poles and zeros in unvoiced speech regions. When the fractal dimension of the speech signal becomes greater than a specified limit (usually in unvoiced speech regions), the covariance matrix and parameter vector are reset to their initiation values. This results in a loss of memory by the algorithm so that, when calculation of the parameters continues, it behaves as if the algorithm had just started. The influence of previously analysed voiced speech regions is thus forgotten, with the result that better tracking of the unvoiced speech region is possible.

We also introduce a procedure to vary the error information according to the gain of the estimated ARMA filter. This, like the fractal dimension estimator, also allows better tracking of the poles and zeros in unvoiced speech regions by choosing a more representative (smaller) value for λ_k. The influence of previously analysed voiced speech regions is thus forgotten by shortening the memory with the use of a smaller forgetting factor. The result is quicker convergence in unvoiced speech. We can summarise the features of the FWRLS as:

Stellenbosch University https://scholar.sun.ac.za

Input estimation: If $\lambda_k < \lambda_0$ (Pulse input)

$$u_k^w = 0$$
$$u_k^p = y_k - \phi_k^t \hat{\theta}_{k-1}$$

If $\lambda_k \geq \lambda_0$ (White noise input)

$$u_k^w = \alpha_k (1 - \phi_k^t K_k)$$
$$u_k^p = 0$$

Variable forgetting factor: $0 < \lambda_k < 1$

Constant Fractal limit: If $(FD_k > F_{limit})$ & $(FD_{k-1} \leq F_{limit})$

with $1 < F_{limit} < 2$ the fractal limit imposed on the recursion and FD_k, the fractal dimension of the speech signal at time instant k.

Error information: Variable according to $E_k = |e_k|^2 + 7 \times G_k$

and with $P_0 = \sigma \mathbf{I}$ the gain of the estimated ARMA filter corresponds to the standard deviation of the estimated white noise input signal.

4.9. CONCLUSION

The initiation of certain parameters influence the rate of convergence, tracking capability and final estimation error (at convergence) of the WRLS-VFF algorithm.

- For quick convergence after starting the algorithm, the covariance can be initiated with $P_0 = \sigma \mathbf{I}$, where σ has a large positive value.
- To improve the convergence rate in unvoiced regions the error information is selected as: $E_k = |e_k|^2 + 7 \times G_k$.
- Tracking over voiced/unvoiced boundaries can be increased by determining these boundaries with a fractal dimension estimator. The covariance matrix is then initiated to allow the fastest possible tracking.

Stellenbosch University https://scholar.sun.ac.za

Chapter 5

Evaluation

5.1. INTRODUCTION

In this chapter we compare the performance of the WRLS-VFF algorithm to other sequential algorithms. The easiest way to do this is to judge the performance of the algorithms on synthetic speech. Another reason for using synthetic speech is that we can control the formant frequencies, bandwidths and excitation signal. We shall use the synthesizer proposed by Klatt [24] which was later refined by Pinto and Childers [25].

5.2. SYNTHESIZING SPEECH

There is a digital formant synthesizer at the heart of every speech synthesizer. The basic building block of a formant synthesizer is a digital resonator. The relationship between the input and output of a resonator is usually defined by the formant frequency F, and the resonance bandwidth B (Pinto and Childers [25], Gold and Rabiner [23]).

$$A_k(z) = \frac{1 - 2r_{pk}\cos(\theta_{pk}) + r_{pk}^2}{1 - 2r_{pk}\cos(\theta_{pk})z^{-1} + r_{pk}^2 z^{-2}}$$

with $r_{pk} = e^{-\frac{\pi B_{pk}}{F_s}}$ and $\theta_{pk} = \frac{2\pi F_{pk}}{F_s}$.

F_{pk} and B_{pk} are the resonant frequency and bandwidth in Hz respectively, of the k'th resonator. F_s is the sampling frequency in Hz.

Anti-formants (zeros) can be generated by an anti-resonator which is the mirror image of the formant resonator:

$$B_k(z) = \frac{1 - 2r_{zk}\cos(\theta_{zk})z^{-1} + r_{zk}^2 z^{-2}}{1 - 2r_{zk}\cos(\theta_{zk}) + r_{zk}^2}$$

Stellenbosch University https://scholar.sun.ac.za

with $r_{zk} = e^{-\frac{\pi B_{zk}}{F_s}}$ and $\theta_{zk} = \frac{2\pi F_{zk}}{F_s}$.

F_{zk} and B_{zk} are the resonant frequency and bandwidth in Hz respectively, of the k'th resonator. F_s is again the sampling frequency in Hz.

To generate synthetic speech with multiple formants (resonance frequencies), the resonators are connected in cascade (Klatt [24]):

$$H(z) = \prod_{k=1}^{N} A_k(z)$$

N is the number of resonant frequencies.

To generate speech signals with both formants and anti-formants, connect the anti-formant resonators in cascade with the formant resonators.

$$H(z) = \prod_{k=1}^{N} A_k(z) \prod_{k=1}^{M} B_k(z)$$

Here M is the number of anti-resonant frequencies.

During the course of this project, we focus our attention on the study of two vowels (/i/ and /a/), two nasal sounds (/m/ and /m/-/i/) and two diphthongs (/i/-/a/ and /a/-/i/). The input to the resonator is a pulse train with a period of 10ms. The sampling frequency used for the synthetic speech is 10000Hz. Table 5.2 shows the formant (and anti-formant) frequencies and bandwidths that are used to generate each of the synthetic speech signals.

	Poles						Zeros	
Sound	f_{p1}	f_{p2}	f_{p3}	$ß_{p1}$	$ß_{p2}$	$ß_{p3}$	f_{z1}	$ß_{z1}$
syn-/i/	270	2290	3010	75	95	120	-	-
syn-/a/	730	1090	2240	60	85	110	-	-
syn-/m/	390	1250	2150	60	150	200	780	80

Table 5.2 Frequencies and Bandwidths used for generating the synthetic speech signals /i/, /a/ and /m/.

The synthetic speech signals for /i/ (syn_i),/a/ (syn_a) and /m/ (syn_m) are shown in figures 5.13, 5.15 and 5.17. The frequency responses of the synthetic speech signals for /i/, /a/ and /m/ are shown in figures 5.14, 5.16 and 5.18.

Stellenbosch University https://scholar.sun.ac.za

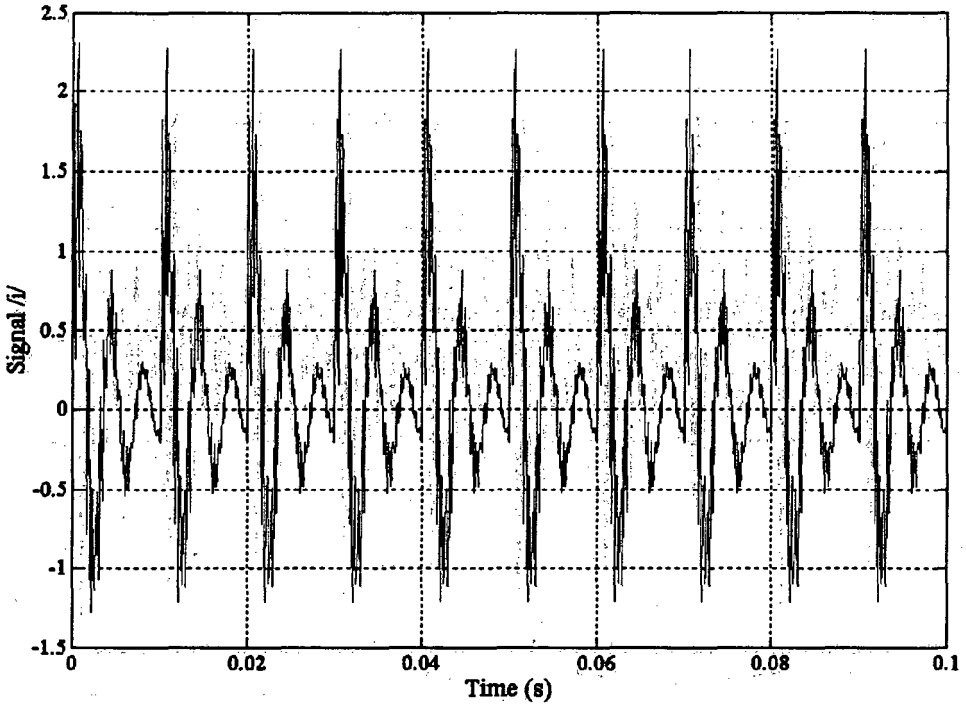


Figure 5.13 Synthetic speech signal /i/ (syn_i).

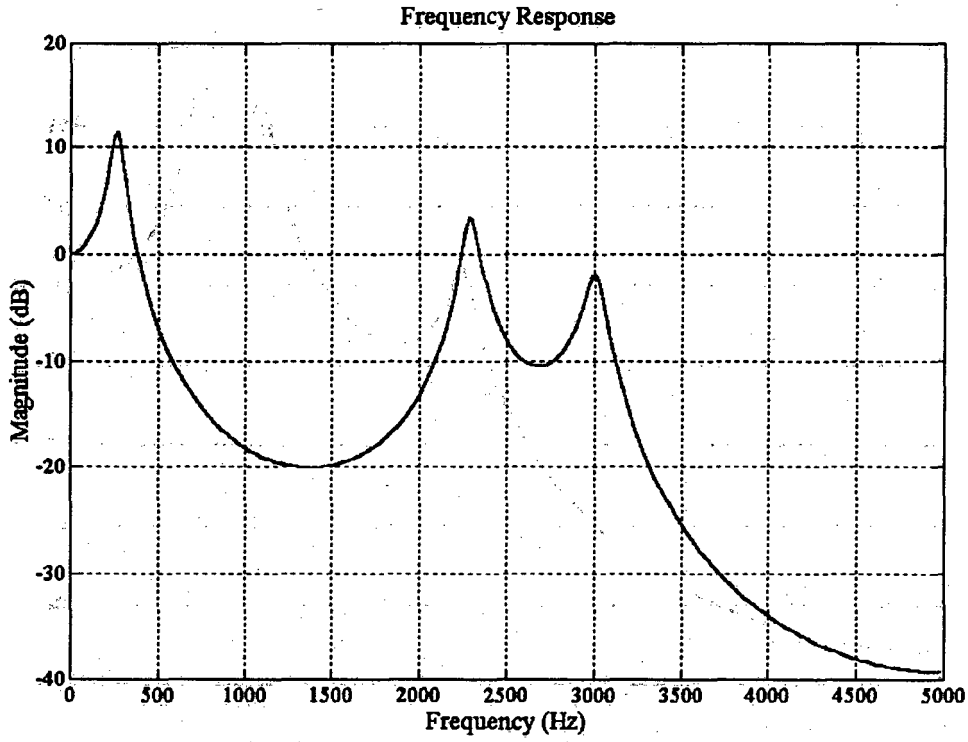


Figure 5.14 Frequency response of synthetic speech signal /i/ (syn_i).

Stellenbosch University https://scholar.sun.ac.za

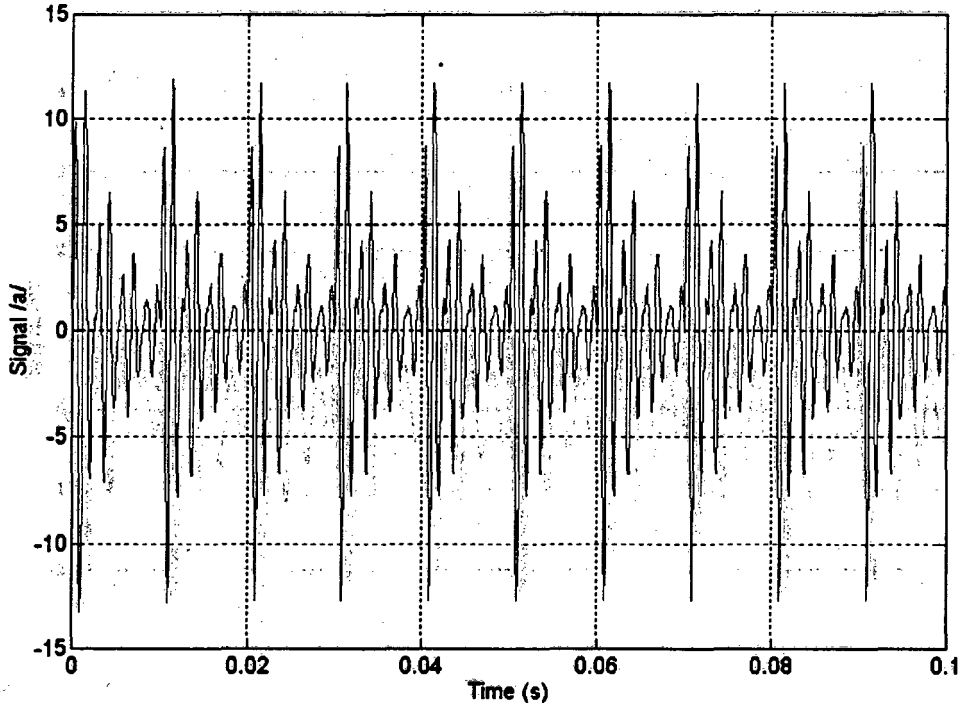


Figure 5.15 Synthetic speech signal /a/ (syn_a).

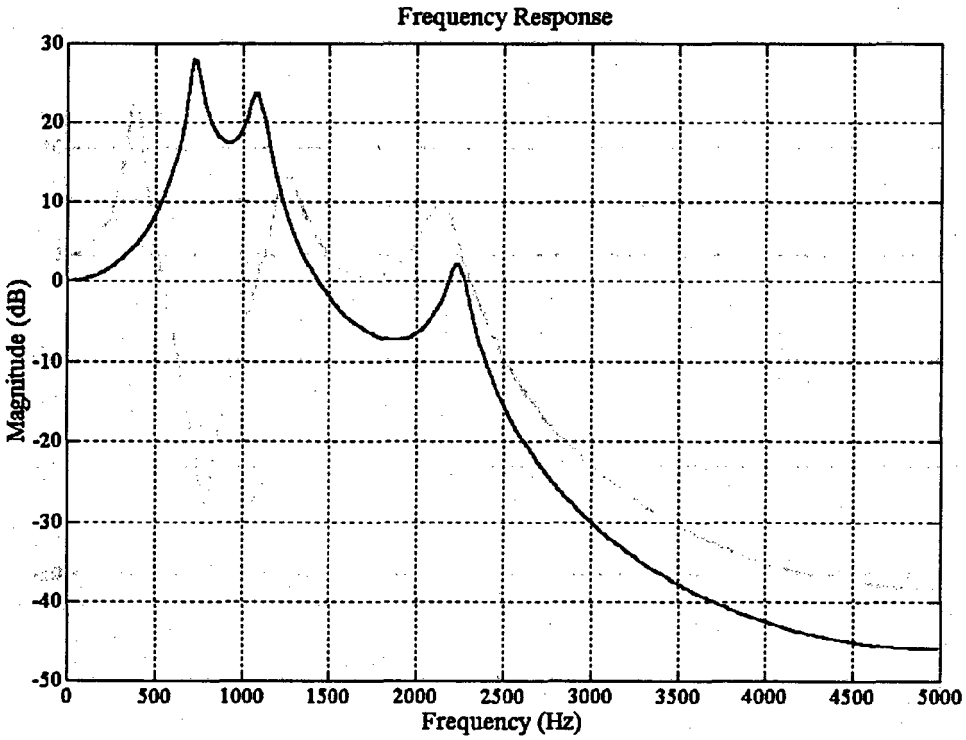


Figure 5.16 Frequency response of synthetic speech signal /a/ (syn_a).

Stellenbosch University https://scholar.sun.ac.za

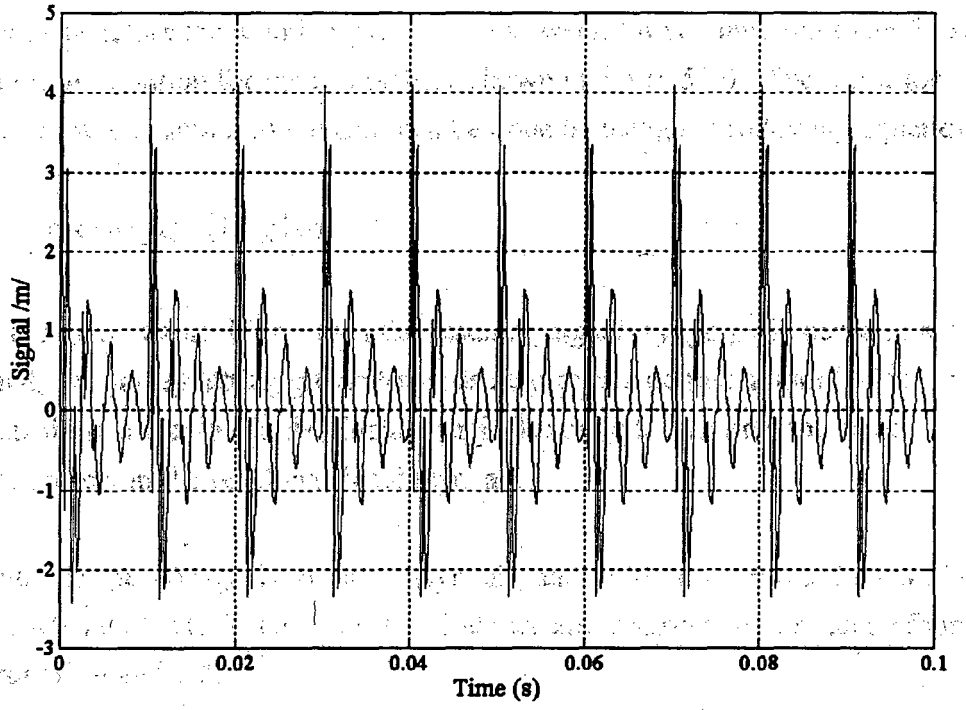


Figure 5.17 Synthetic speech signal /m/ (syn_m).

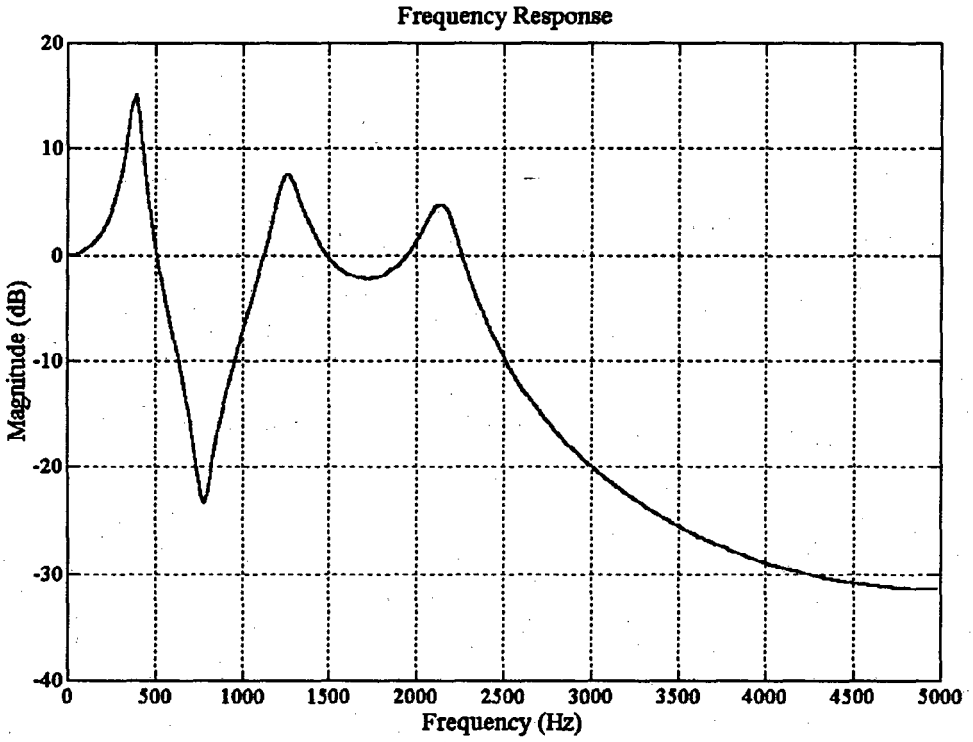


Figure 5.18 Frequency response of synthetic speech signal /m/ (syn_m).

Stellenbosch University https://scholar.sun.ac.za

In order to generate the diphthong /a/-/i/, the vowel /a/ was converted to an /i/ by using a linear variation for the formants as shown in figure 5.20. The transition from one synthetic sound to the next can be done by using the following equation:

$$syn\_ai = g \times syn\_a + (1-g)syn\_i$$

Here g is a gain factor. If g=1 then the resulting signal (syn_ai) will consist only of the signal, syn_a. By letting g fade towards zero over a certain amount of time we can control the transition to the signal, syn_i. When g=0 then only the signal, syn_i, is present in the resulting signal (syn_ai).

The synthetic speech signals for /a/-/i/ (syn_ai), and /m/-/i/ (syn_mi) are shown in figures 5.19, and 5.21. Their respective 3-dimensional spectrograms can be found in figures 5.20, and 5.22.

Stellenbosch University https://scholar.sun.ac.za

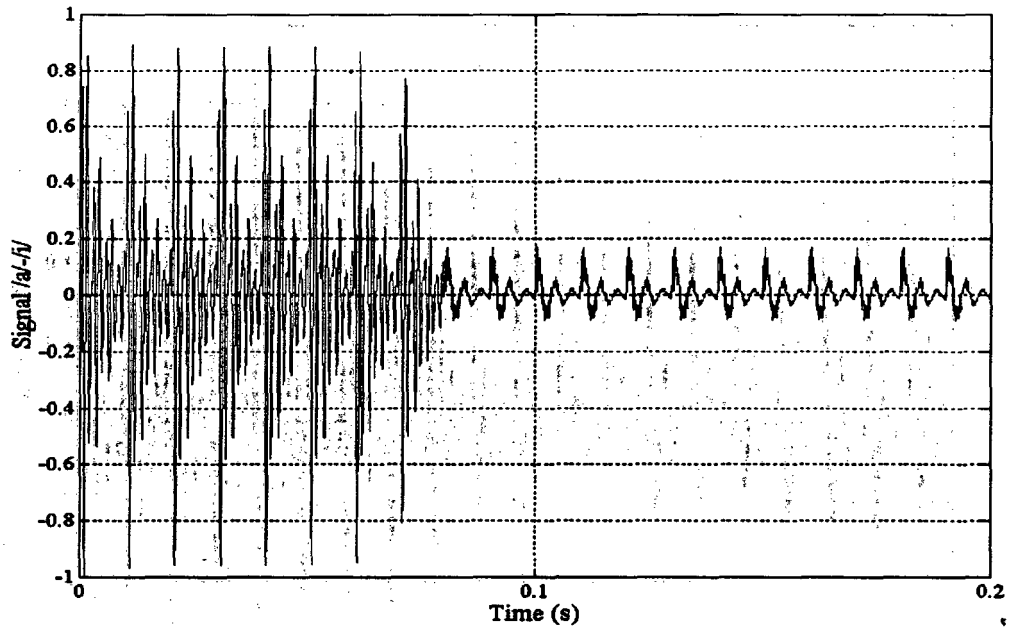


Figure 5.19 Synthetic speech signal /a/-/i/ (syn_ai).

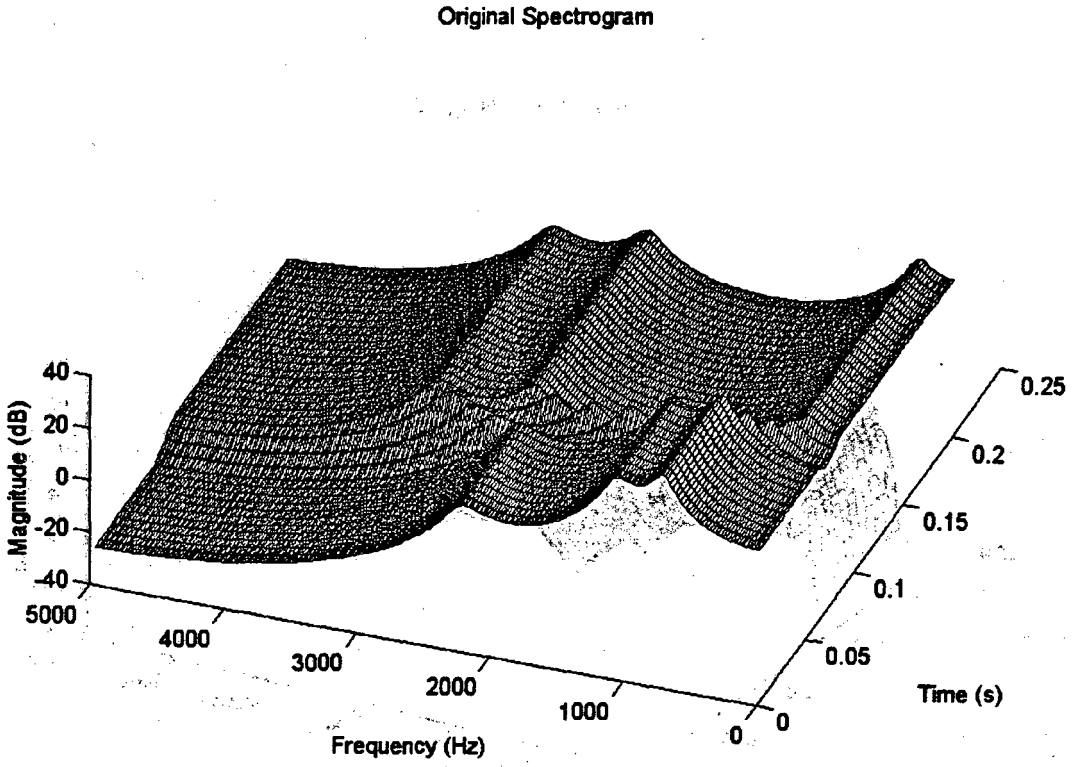


Figure 5.20 Original 3D Spectrogram for the synthetic speech signal /a/-/i/ (syn_ai).

Stellenbosch University https://scholar.sun.ac.za

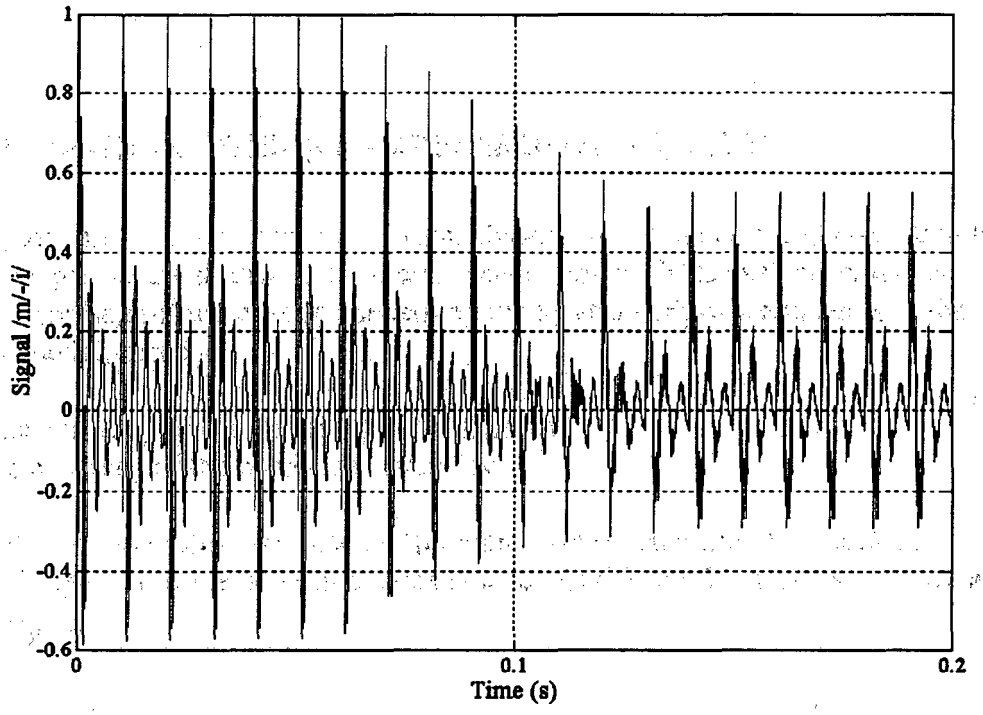


Figure 5.21 Synthetic speech signal /m/-/i/ (syn_mi).

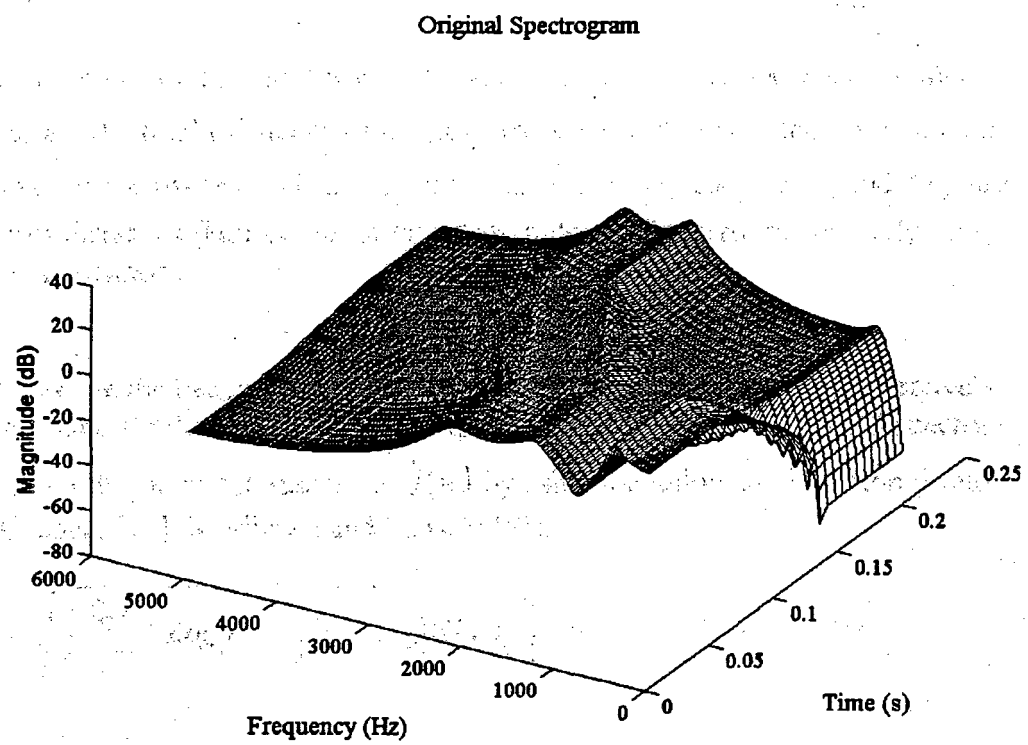


Figure 5.22 Original 3D Spectrogram the synthetic speech signal /m/-/i/ (syn_mi).

Stellenbosch University https://scholar.sun.ac.za

5.3. EVALUATION CRITERIA & EXPERIMENTAL RESULTS

In this section the performance of several algorithms is evaluated and compared by using the previously discussed synthetic speech signals. This evaluation will be made according to two criteria concerning the location of poles and zeros in the speech model, namely:

(1) The accuracy of the frequency estimation and
(2) the accuracy of the bandwidth estimation.

The following definitions are used to (quantitatively) determine the frequency estimation errors of the poles and zeros respectively: (modified from Morikawa and Fujisaki [7])

$$E_p = \sum_{i=1}^{p} \frac{\Delta F_{pi}}{F_s/2} \qquad 5.1$$

$$E_z = \sum_{i=1}^{q} \frac{\Delta F_{zi}}{F_s/2} \qquad 5.2$$

where p and q are the number of poles and zeros of the ARMA speech model. E_p and E_z are the total frequency estimation errors associated with the estimation of the poles and zeros respectively. F_s is the sampling frequency. ΔF_{pi} and ΔF_{zi} are the error distances (Hz) of the estimated i'th pole and i'th zero, respectively, from their real locations.

To determine the bandwidth estimation error for the poles and zeros respectively we can calculate the root mean square (RMS) error between the original spectrum $H(w)$ and the estimated spectrum $\hat{H}(w)$ by using the following definition (Pinto and Childers [25], Morikawa and Fujisaki [9]):

$$E_{rms} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}\left[10log\left|H(w)\right|^2 - 10log\left|\hat{H}(w)\right|^2\right]^2} \qquad 5.3$$

Stellenbosch University https://scholar.sun.ac.za

5.3.1. Pole-zero detection and tracking

The synthetic speech signal of /m/ (syn_m) was analysed by means of various algorithms in order to compare their ability to accurately *detect* the locations of poles and zeros. The results are shown in table 5.3.

Because syn_m is a stationary signal, a value close to unity (0.99) for λ was selected in the case of the CWRLS. The values for E_0, in the MWRLS and WRLS-VFF algorithms, were determined experimentally, as was mentioned earlier. The upper threshold/limit for the fractal dimension was chosen as 1.5 in all the tests on synthetic speech. The fractal method itself will have no influence on the estimation of the poles and zeros unless the fractal dimension of the speech signal increases above this threshold. This will only happen in silent parts of a real speech signal or when a sudden transient from high energy to low energy occurs. The effectiveness of using the fractal dimension estimator will be pointed out when the tests on syn_ai are discussed, where the higher energy signal, /a/, is followed by the lower energy signal, /i/.

	SEARMA	CWRLS	MWRLS	WRLS-VFF	FWRLS
λ	1.0	0.99	Variable	Variable	Variable
λ_{min}	-	-	0.93	$\frac{2(p+q)-1}{2(p+q)}$	$\frac{2(p+q)-1}{2(p+q)}$
λ_0	-	-	$\lambda_{min}+0.01$	$\lambda_{min}+0.01$	$\lambda_{min}+0.01$
E_0	-	-	0.075	0.2	Variable
F_l	-	-	-	-	1.5
E_p (%)	0.5390	0.8847	0.6486	0.5160	0.2750
E_q (%)	0.7430	1.4582	0.8984	0.5078	0.2109
E_{rms} (dB)	2.4865	2.2840	1.9931	1.9295	1.9936

Table 5.3 Comparison of different sequential algorithms on the detection of pole and zero locations in the synthetic speech signal /m/.

From table 5.3 the following conclusions are drawn:

- The FWRLS obtains the best results in detecting and tracking the pole and zero frequencies. The estimation errors associated with the pole and zero frequencies are less than half of those produced by the next best method, namely the WRLS-VFF.

- The spectral distance error (associated with the bandwidths of the poles and zeros) is slightly less (0.07%) for the WRLS-VFF than for the FWRLS algorithm. In speech processing this is of less importance than the correct estimation of the formant/anti-formant frequencies.

Stellenbosch University https://scholar.sun.ac.za

- Notice that the SEARMA method does exceptionally well. This is because the signal under consideration is stationary over the whole analysis interval.

In order to compare the capability of various algorithms to accurately *track* the variations of poles and zeros during a speech utterance the synthetic speech signal of /m/-/i/ (syn_mi) is analysed. The results are shown in table 5.4.

Since the exact contours of each of the test signals are known, the performance of the algorithms can easily be verified by comparing the test results with the original formant/antiformant tracks.

	SEARMA	CWRLS	MWRLS	WRLS-VFF	FWRLS
λ	1.0	0.96	Variable	Variable	Variable
λ_{min}	-	-	.93	$\frac{2(p+q)-1}{2(p+q)}$	$\frac{2(p+q)-1}{2(p+q)}$
λ_0	-	-	$\lambda_{min}+0.01$	$\lambda_{min}+0.01$	$\lambda_{min}+0.01$
E_0	-	-	0.001	0.2	Variable
F_l	-	-	-	-	1.5
E_p(%)	15.48	5.4523	3.0610	10.9516	2.8899
E_q(%)	3.1332	3.0298	5.3604	2.2779	1.5262
E_{rms}(dB)	10.0054	9.7527	9.9474	9.7084	9.7711

Table 5.4 Comparison of different sequential algorithms on the detection of pole and zero locations in the synthetic speech signal /m/-/i/.

Because syn_mi is not a stationary signal, $\lambda = 0.96$ is selected in the CWRLS method. In computing the frequency error for the deviation from the original zero tracks, only the parts where zeros really exist are used. Thus the part where only the signal /i/ is present, is not used in the computation of E_q.

From table 5.4 the following conclusions are drawn:

- From the 5 methods tested, the FWRLS fares best again for estimating the pole and zero frequencies. We can see that our proposed method for varying the value of E_0 is very effective. It allows tracking in regions where the WRLS-VFF fails.
- Again the WRLS-VFF shows only a slightly better performance (only 0.07% better than the FWRLS) on the spectral distance error.
- It is clear that the CWRLS and MWRLS achieve better results than the WRLS-VFF, in estimating the pole frequencies. They achieve this by inserting the excess zeros between the two higher frequency poles, in the part of the signal

Stellenbosch University https://scholar.sun.ac.za

where only the all-pole sound (/i/) exists.

- The frequency estimation error for the zeros in the case of the WRLS-VFF is lower than that of the CWRLS and MWRLS methods.

In the last test on synthetic speech, the signal syn_ai is used to show the effectiveness of using the fractal dimension estimator. The analysed signal consists of a high energy region (/a/) followed by a much lower energy part (/i/), which is detected by the fractal dimension estimator.

The results are summarised in table 5.5. Notice that table 5.5 contains an extra column to demonstrate the effect of not using the fractal dimension estimator when there is a transient from high to lower energy speech.

	SEARMA	CWRLS	MWRLS	WRLS-VFF	FWRLS	FWRLS without the fractal dimension estimator
λ	1.0	0.93	Variable	Variable	Variable	Variable
λ_{min}	-	-	0.93	$\frac{2(p+q)-1}{2(p+q)}$	$\frac{2(p+q)-1}{2(p+q)}$	$\frac{2(p+q)-1}{2(p+q)}$
λ_0	-	-	-	$\lambda_{min}+0.01$	$\lambda_{min}+0.01$	$\lambda_{min}+0.01$
E_0	-	-	0.01	0.01	Variable	Variable
F_1	-	-	-	-	1.5	-
E_p(%)	29.6129	29.0759	29.0794	25.9842	3.6042	26.7368
E_q(%)	-	-	-	-	-	-
E_{rms}(dB)	9.3437	6.9477	7.5662	8.1077	2.6630	8.0698

Table 5.5 Comparison of different sequential algorithms on the detection of pole and zero locations in the synthetic speech signal /a/-/i/.

A very low value ($\lambda = 0.93$) is selected for the constant forgetting factor in the CWRLS method. This allows it to follow very fast changes in the non-stationary signal.

Comparing the results in table 5.5 shows the following:

- The frequency error estimation for the FWRLS is more than 7 times lower than that of the next best method, namely the WRLS-VFF.
- Even the spectral distance error is 2.5 times smaller than that of the closest method, namely the CWRLS.

Stellenbosch University https://scholar.sun.ac.za

- Without using the fractal dimension estimator, the error rates fall to almost the same as that of the WRLS-VFF.

We conclude that in all three of the tests on synthetic speech the FWRLS did by far the best in estimating both pole and zero frequencies. The lowest error rates were recorded on stationary signals (E_p = 0.2750% and E_q = 0.2109% on syn_m). The highest error rates were recorded on the non-stationary signal with the transition from high to low energy(E_p = 3.6042% and E_q = 1.5262% on syn_ai).

The WRLS-VFF achieved the next best results. Again the lowest error rates were obtained on the stationary signal(E_p = 0.5160% and E_q = 0.5078% on syn_m). The highest error rates were E_p = 25.984% on syn_ai and E_q = 2.2779% on syn_mi.

When comparing the spectral distances between the original and estimated spectrums, the WRLS-VFF fared best for syn_m and syn_mi, but it was only 0.07% better than the FWRLS. For syn_ai the FWRLS obtained E_{rms} = 2.6630 compared to the E_{rms} = 8.1077 of the WRLS-VFF (a difference of 5.44%).

The estimated 3D-spectrograms for the different methods, tested on syn_mi, are shown in figures 5.23, 5.24, 5.25, 5.26 and 5.27. The pole tracks, estimated by each method tested on syn_mi, are compared to the original pole tracks in figures 5.28, 5.29, 5.30, 5.31 and 5.32.

The estimated 3D-spectrograms for the different methods, tested on syn_ai, are shown in figures 5.33, 5.35, 5.37, 5.39 and 5.41. The pole tracks, estimated by each method tested on syn_ai, are compared to the original pole tracks in figures 5.34, 5.36, 5.38, 5.40 and 5.42.

The zero tracks, estimated by each method tested on syn_mi, are compared to the original zero tracks in appendix E. Also in appendix E are the spectral distance error (E_{rms}) graphs for both syn_mi and syn_ai, analysed with all five methods.

Stellenbosch University https://scholar.sun.ac.za

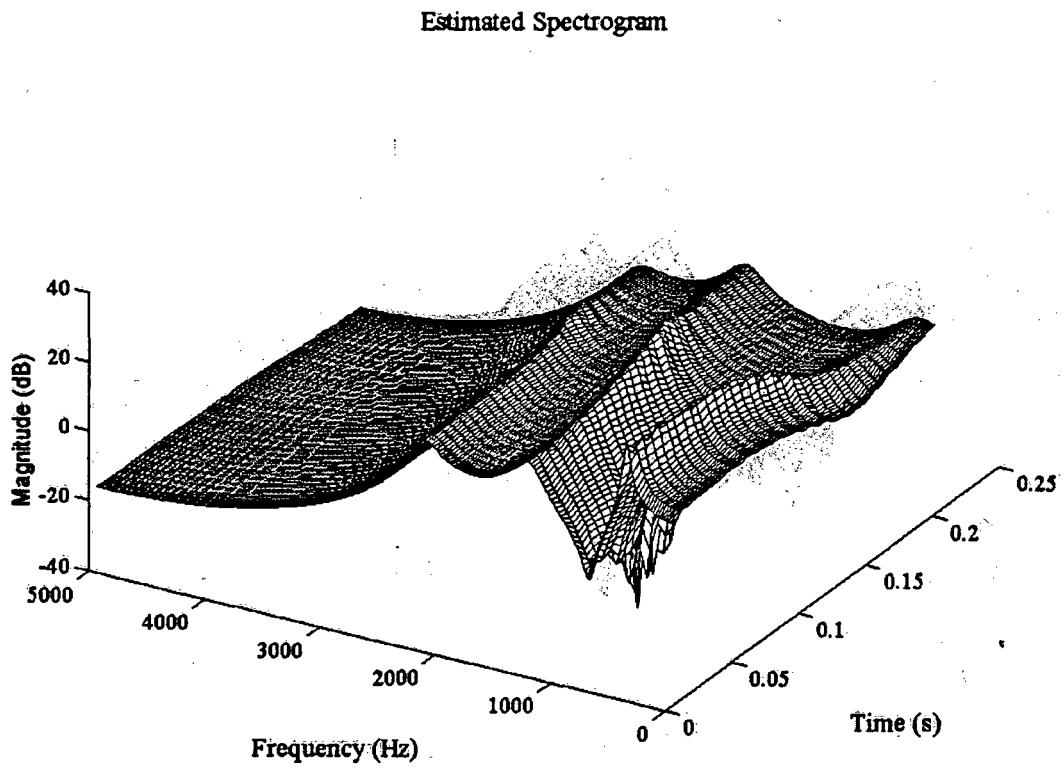


Figure 5.23 Estimated 3D Spectrogram of the synthetic speech signal /m/-/i/ (syn_mi) (SEARMA).

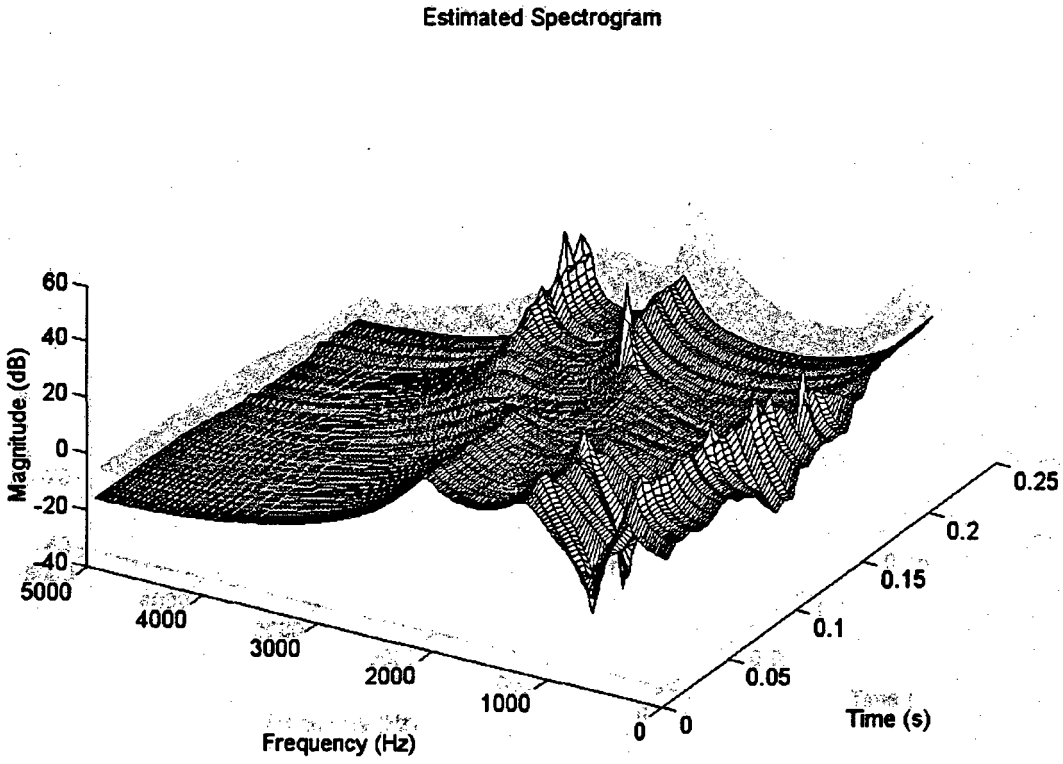


Figure 5.24 Estimated 3D Spectrogram of the synthetic speech signal /m/-/i/ (syn_mi) (CWRLS).

Stellenbosch University https://scholar.sun.ac.za

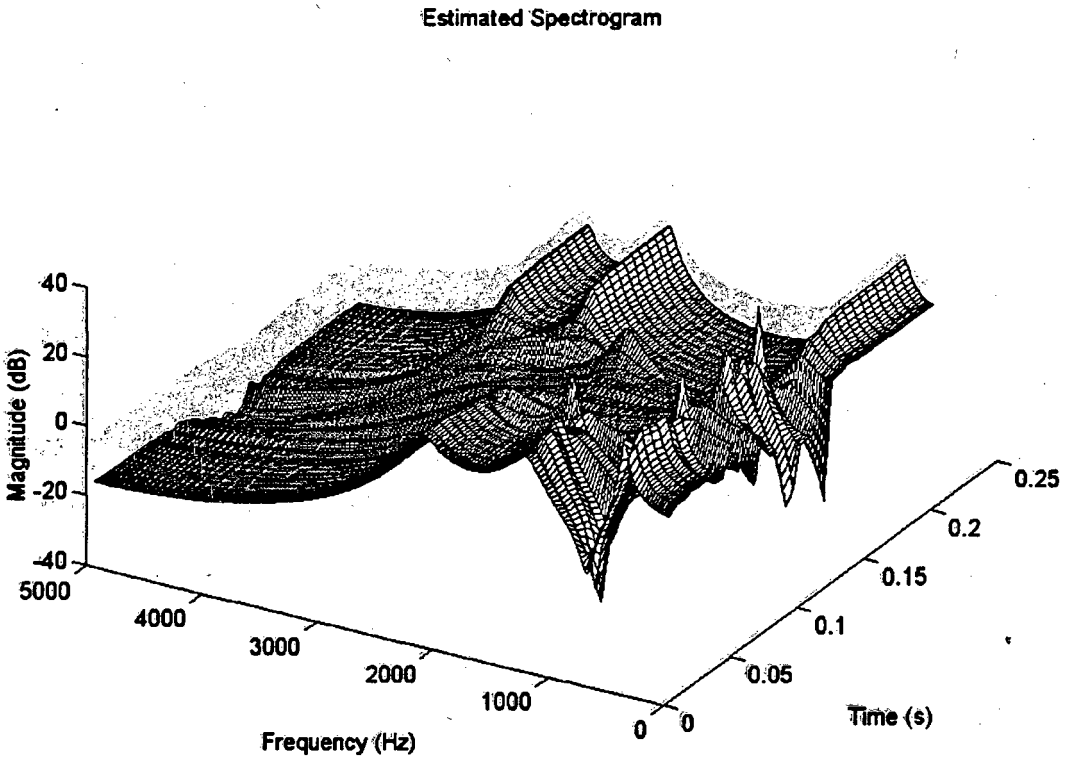


Figure 5.25 Estimated 3D Spectrogram of the synthetic speech signal /m/-/i/ (syn_mi) (MWRLS).

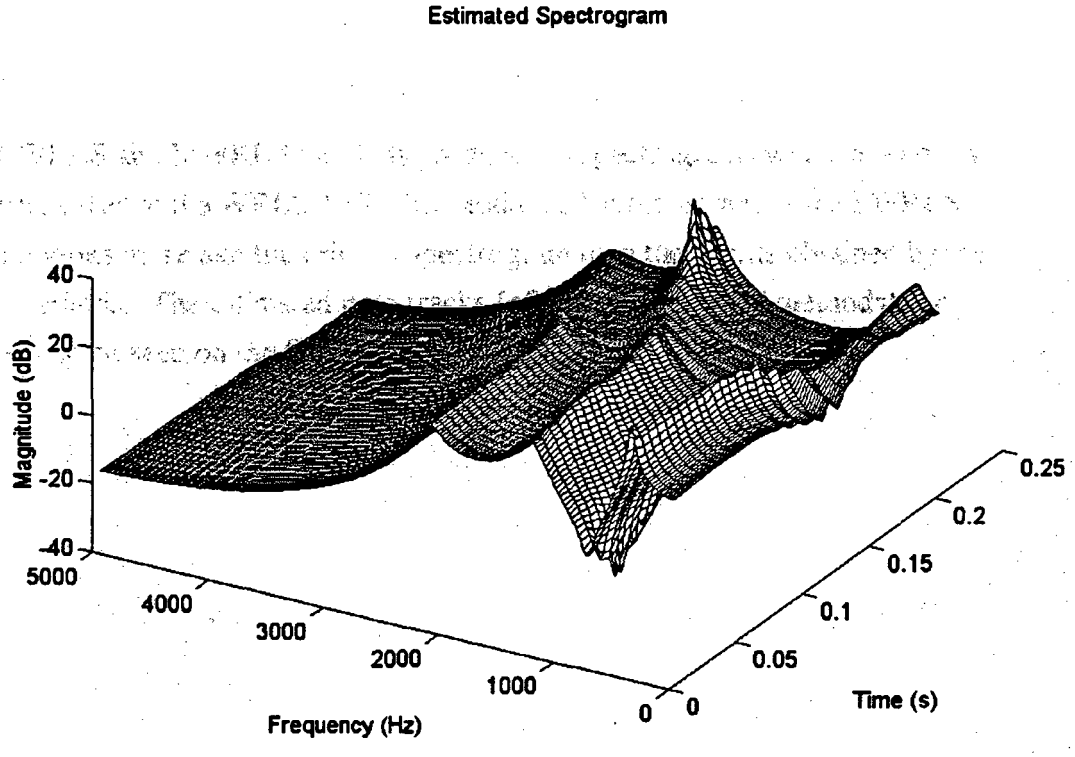


Figure 5.26 Estimated 3D Spectrogram of the synthetic speech signal /m/-/i/ (syn_mi) (WRLS-VFF).

Stellenbosch University https://scholar.sun.ac.za

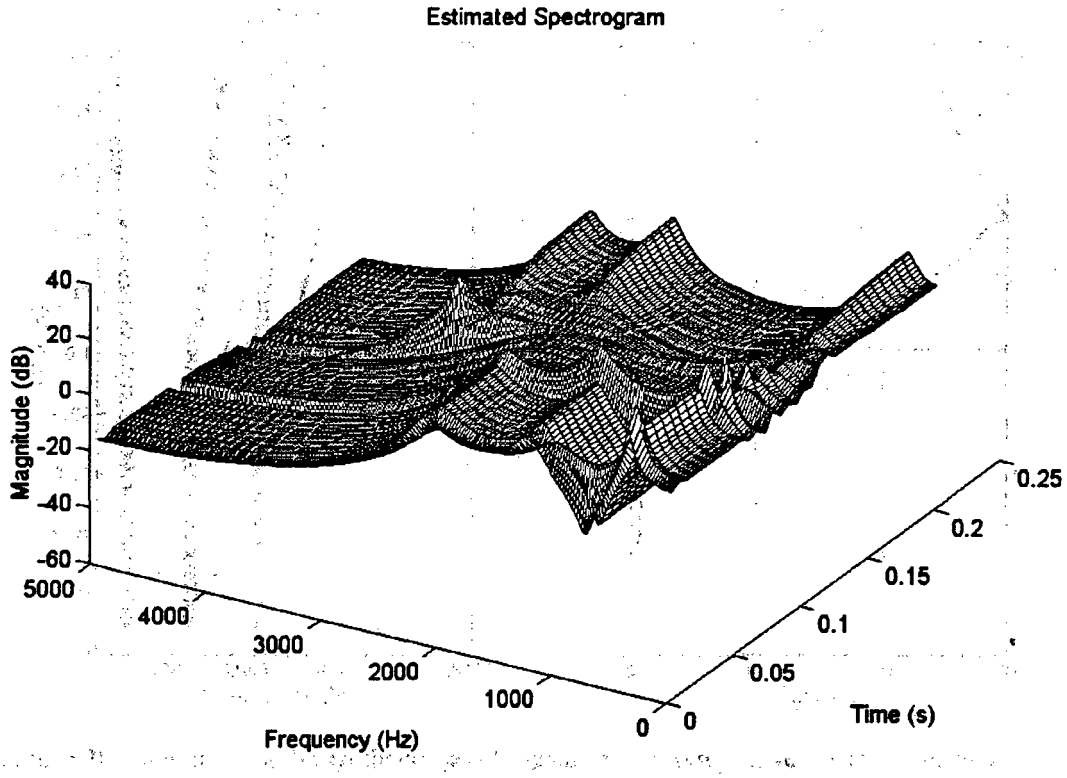


Figure 5.27 Estimated 3D Spectrogram of the synthetic speech signal /m/-/i/ (syn_mi) (FWRLS).

The CWRLS and MWRLS methods produce 3D spectrograms which are not as smooth as that of the WRLS-VFF. The estimated spectrogram of the FWRLS method looks more like the original spectrogram than the results obtained by the other methods. The estimated pole-tracks (of all five sequential methods) for syn_mi can be seen on the following few pages.

Stellenbosch University https://scholar.sun.ac.za

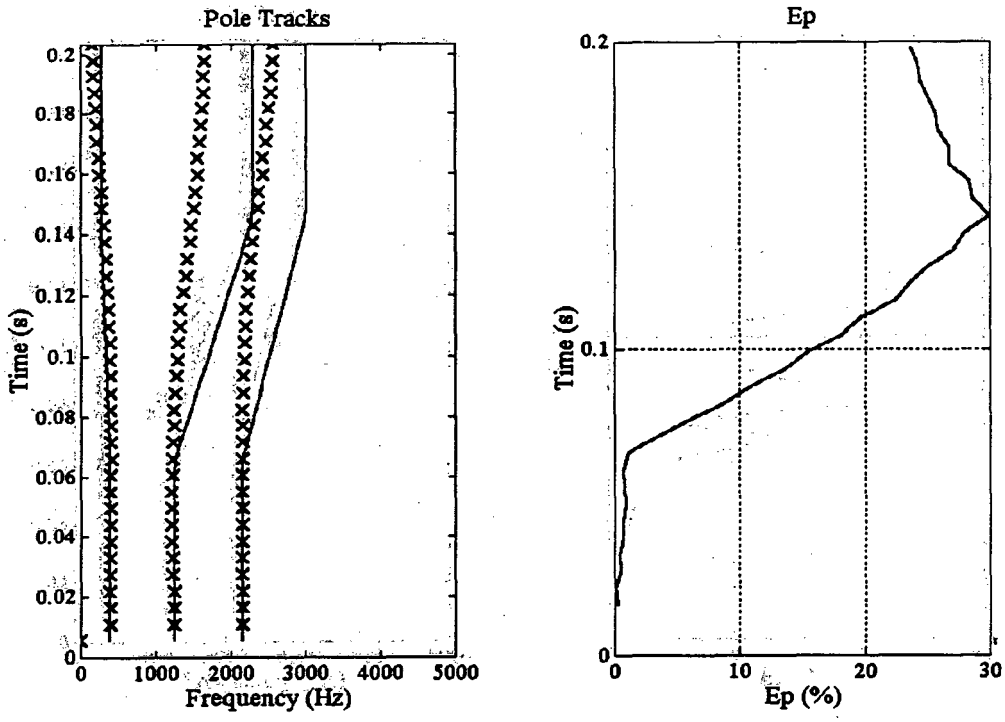


Figure 5.28 Pole tracks of the synthetic speech signal /m/-/i/ (syn_mi) versus the estimated tracks (SEARMA).

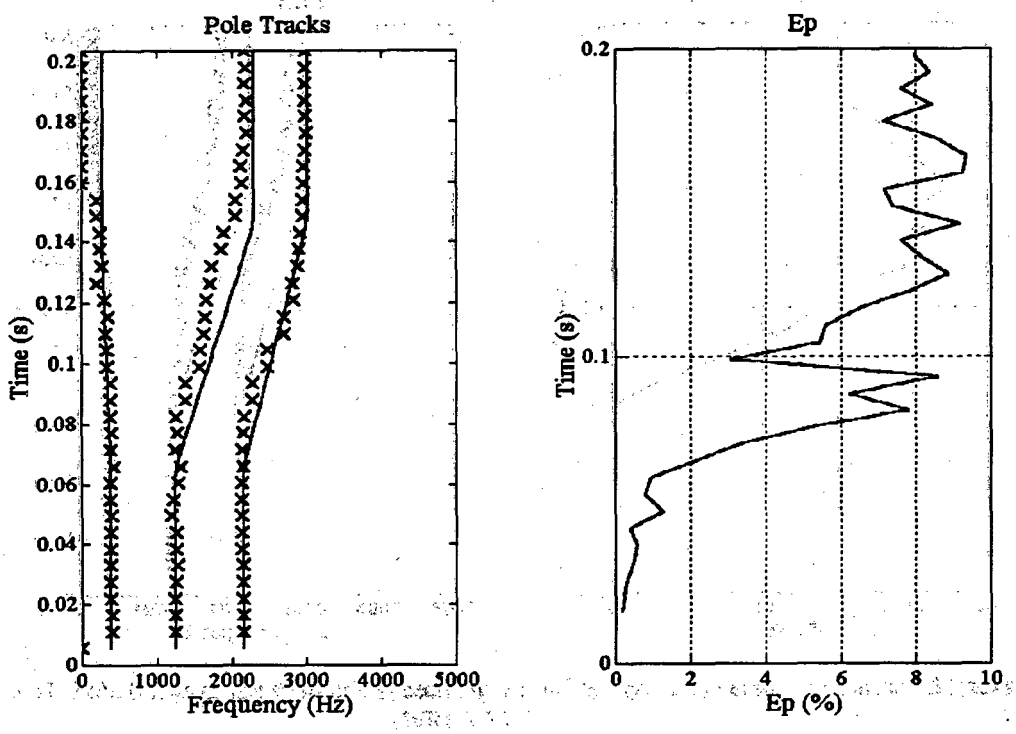


Figure 5.29 Pole tracks of the synthetic speech signal /m/-/i/ (syn_mi) versus the estimated tracks (CWRLS).

Stellenbosch University https://scholar.sun.ac.za

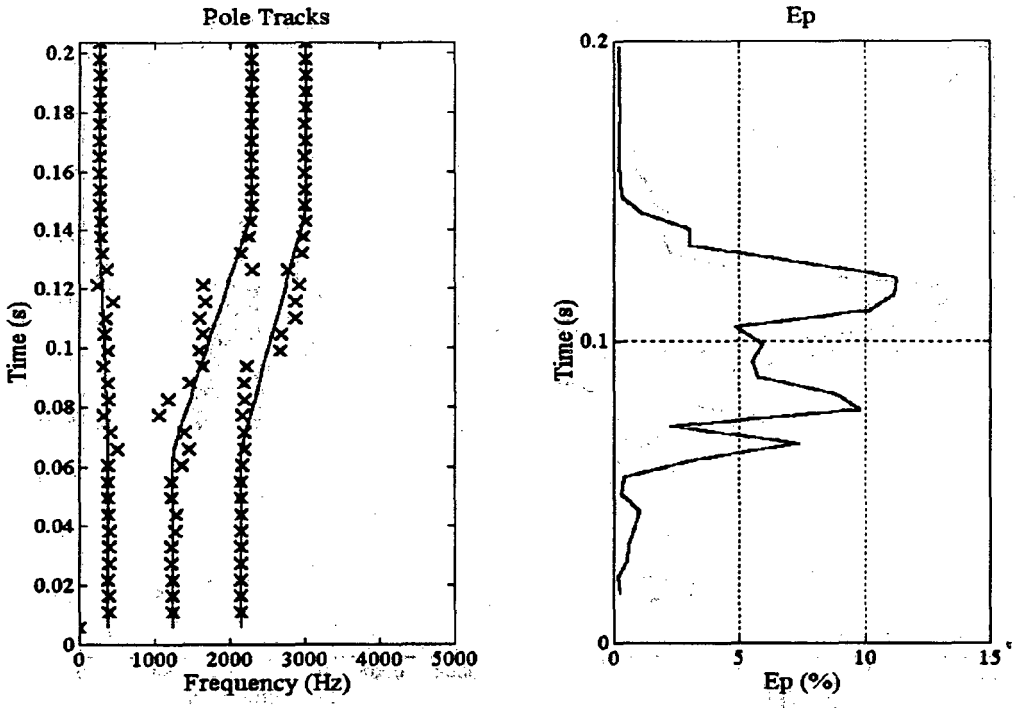


Figure 5.30 Pole tracks of the synthetic speech signal /m/-/i/ (syn_mi) versus the estimated tracks (MWRLS).

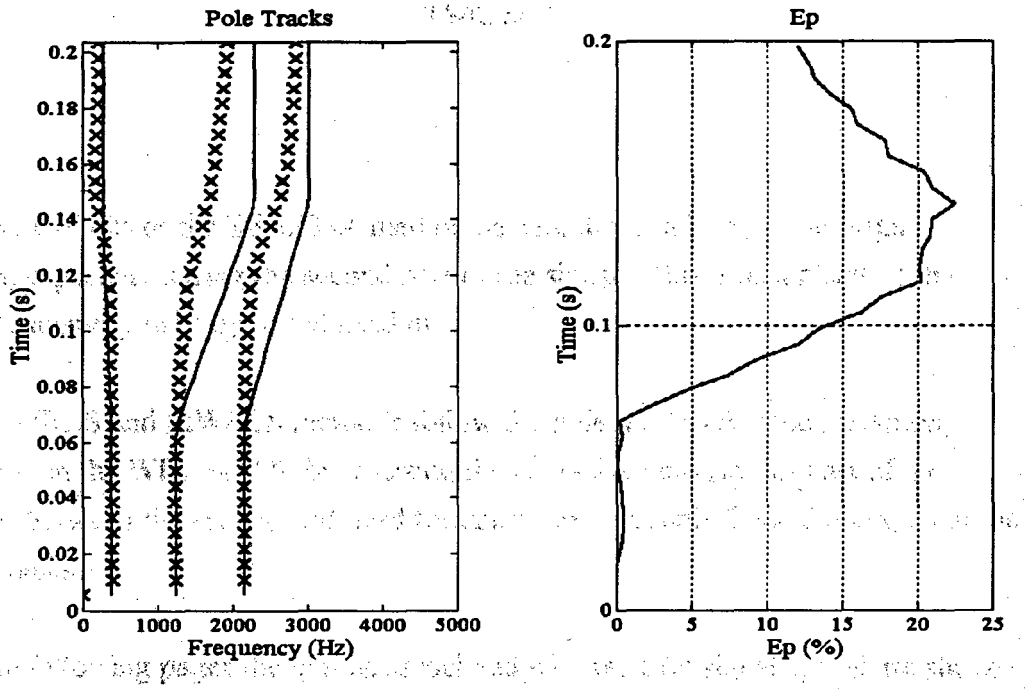


Figure 5.31 Pole tracks of the synthetic speech signal /m/-/i/ (syn_mi) versus the estimated tracks (WRLS-VFF).

Stellenbosch University https://scholar.sun.ac.za

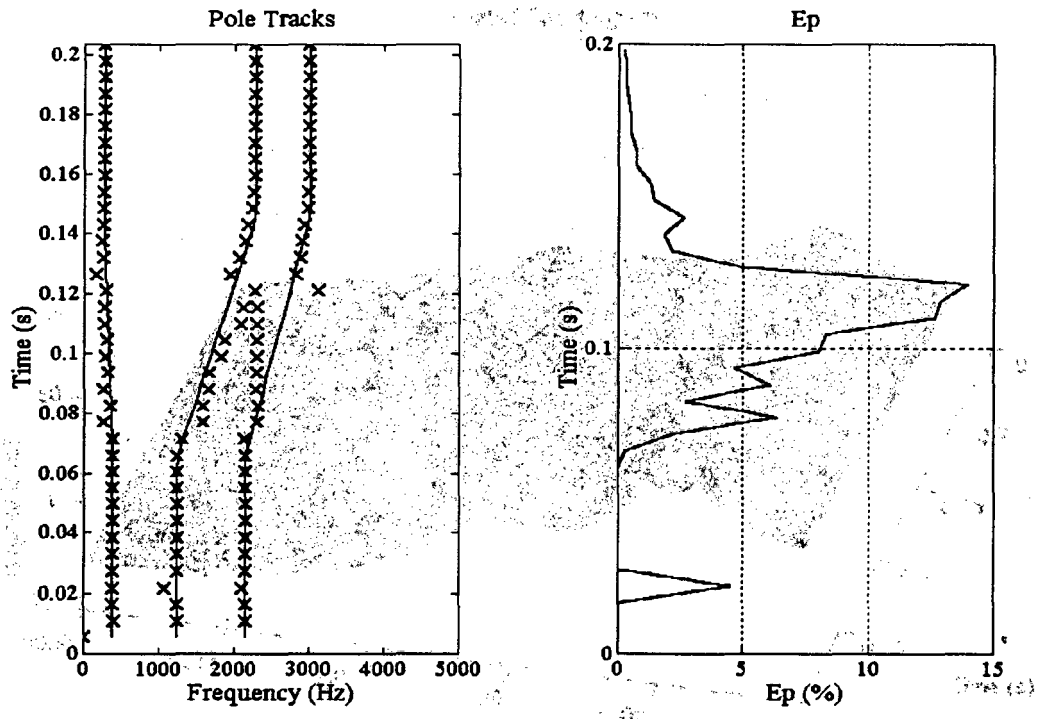


Figure 5.32 Pole tracks of the synthetic speech signal /m/-/i/ (syn_mi) versus the estimated tracks (FWRLS).

The pole tracks of the SEARMA method deviate drastically from the original tracks, especially during the second part of the signal. This is as a result of the infinite memory used by this algorithm.

The CWRLS and MWRLS methods follow the pole tracks after the transition, better than the WRLS-VFF, by inserting the excessive zero (in this part of the signal) between the second and third formants (see appendix E for the graphs of the zero tracks).

On the following pages the graphs associated with tests on signal syn_ai are shown. Note from the pole track graphs how the FWRLS detects the short transition from /a/ to /i/. Immediately after this transition the FWRLS follows the signal perfectly.

Stellenbosch University https://scholar.sun.ac.za

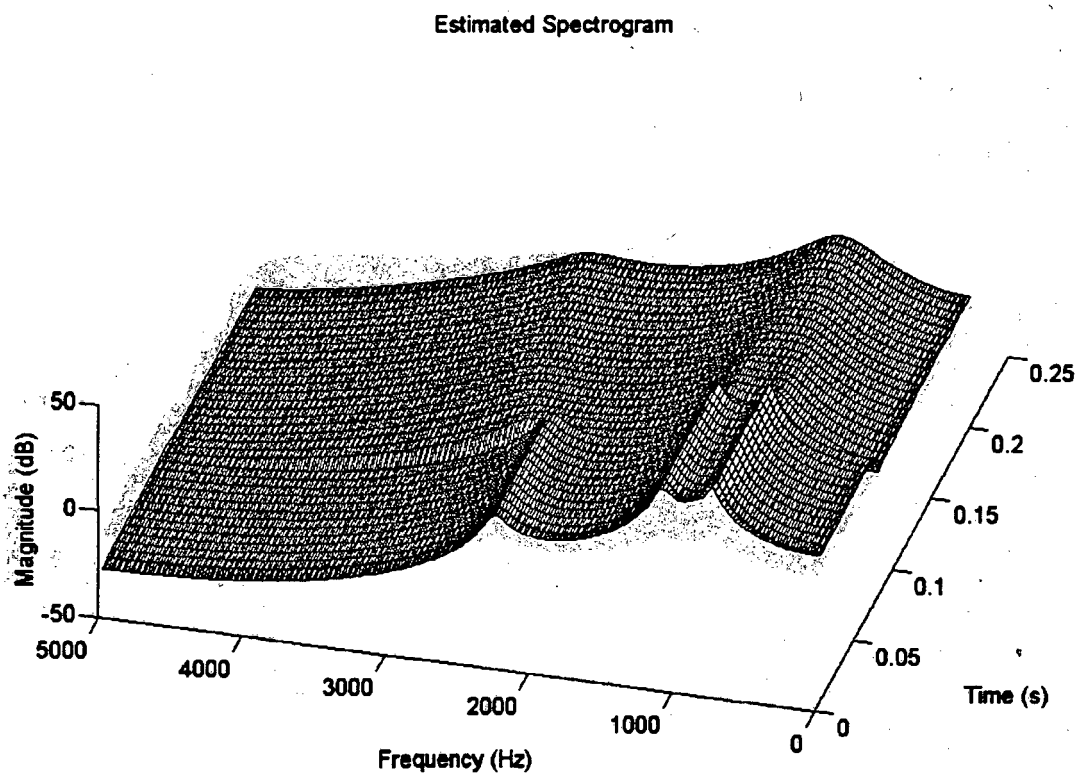


Figure 5.33 Estimated 3D Spectrogram of the synthetic speech signal /a/-/i/ (syn_ai) (SEARMA).

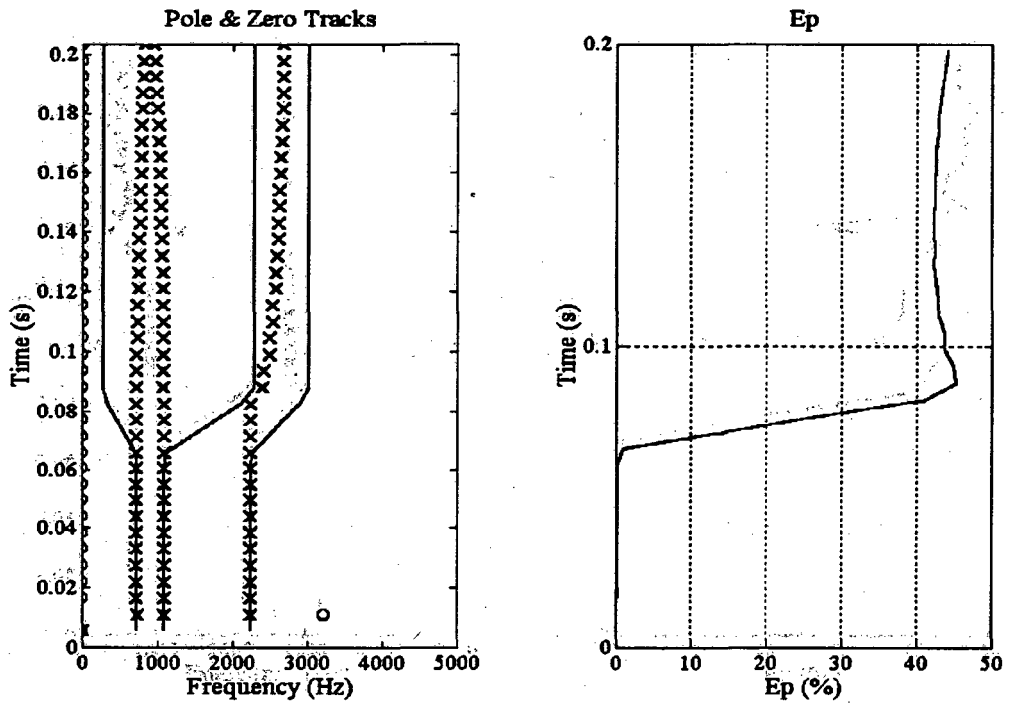


Figure 5.34 Pole-zero tracks of the synthetic speech signal /a/-/i/ (syn_ai) versus the estimated tracks (SEARMA).

Stellenbosch University https://scholar.sun.ac.za

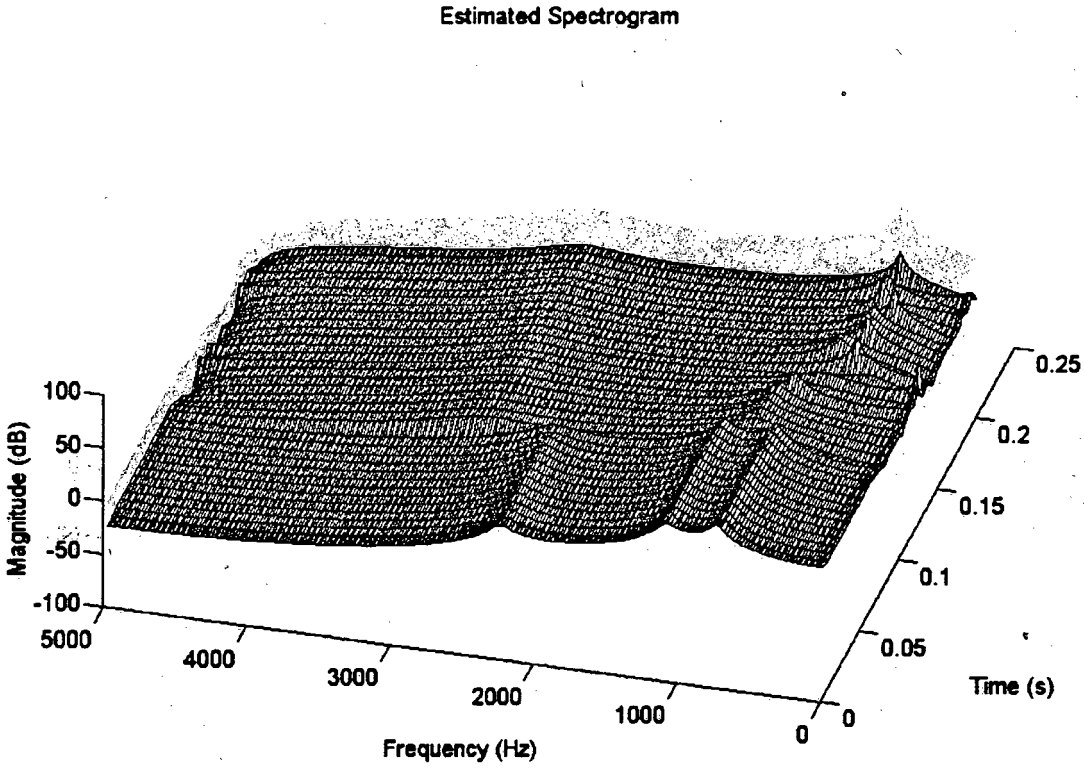


Figure 5.35 Estimated 3D Spectrogram of the synthetic speech signal /a/-/i/ (syn_ai) (CWRLS).

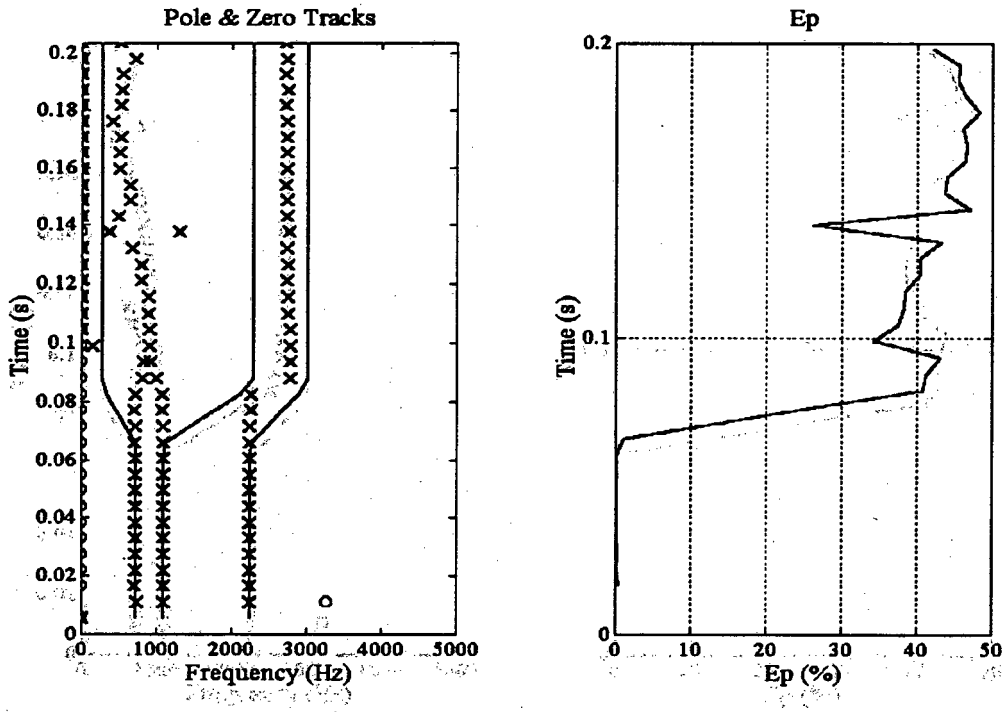


Figure 5.36 Pole-zero tracks of the synthetic speech signal /a/-/i/ (syn_ai) versus the estimated tracks (CWRLS).

Stellenbosch University https://scholar.sun.ac.za

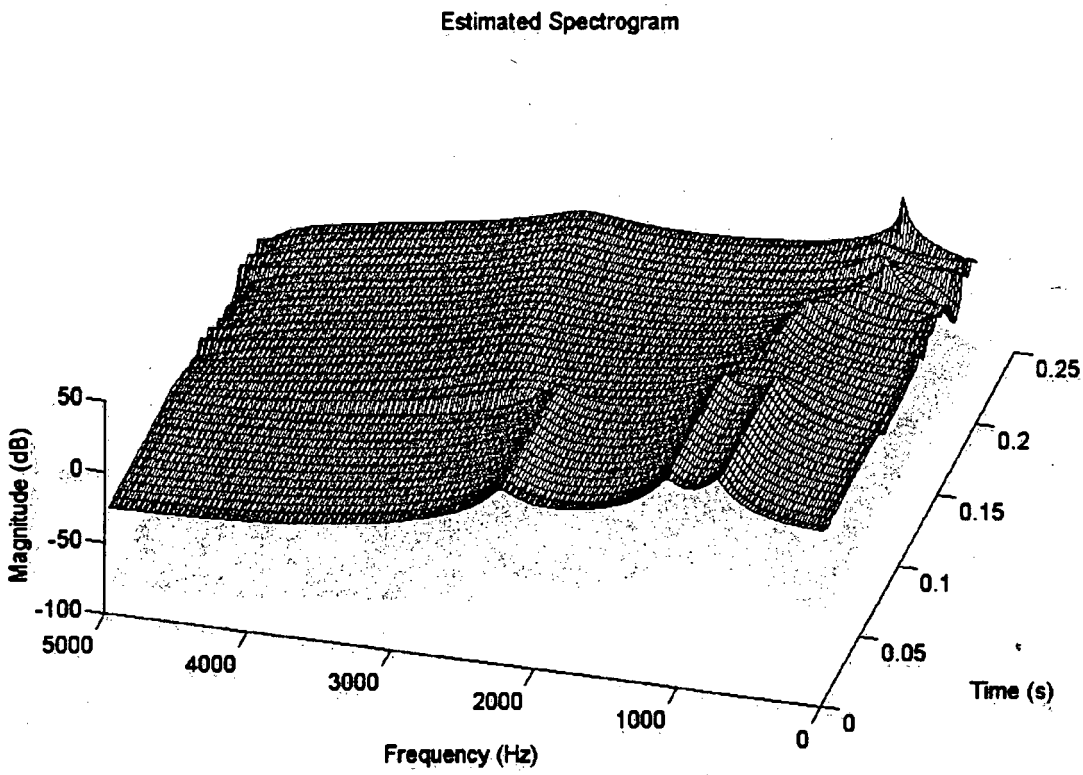


Figure 5.37 Estimated 3D Spectrogram of the synthetic speech signal /a/-/i/ (syn_ai) (MWRLS).

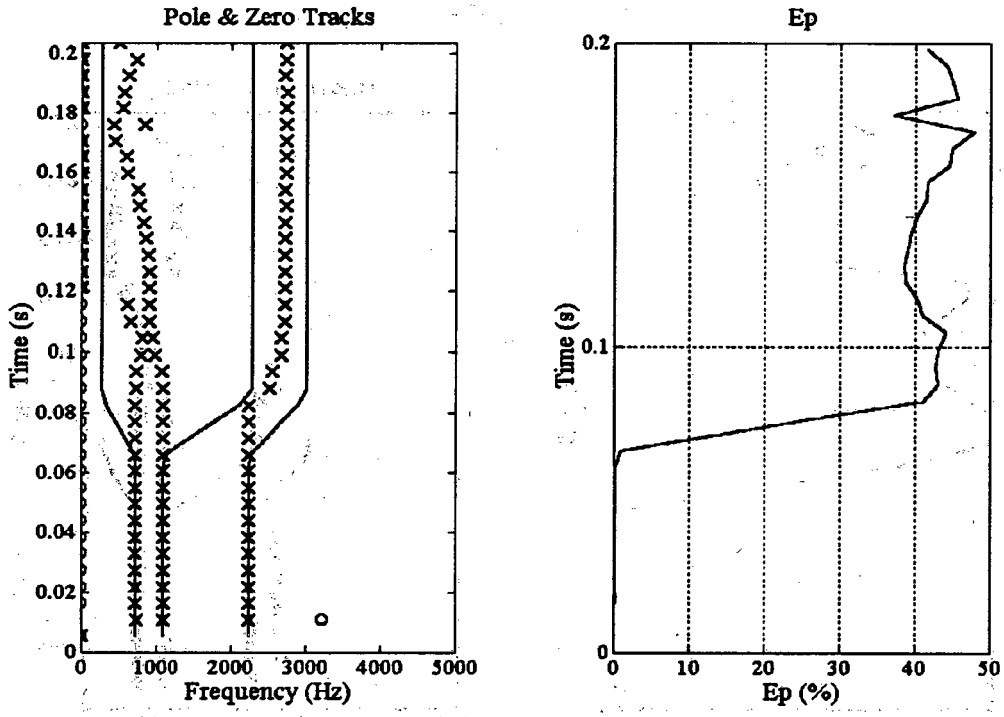


Figure 5.38 Pole-zero tracks of the synthetic speech signal /a/-/i/ (syn_ai) versus the estimated tracks (MWRLS).

Stellenbosch University https://scholar.sun.ac.za

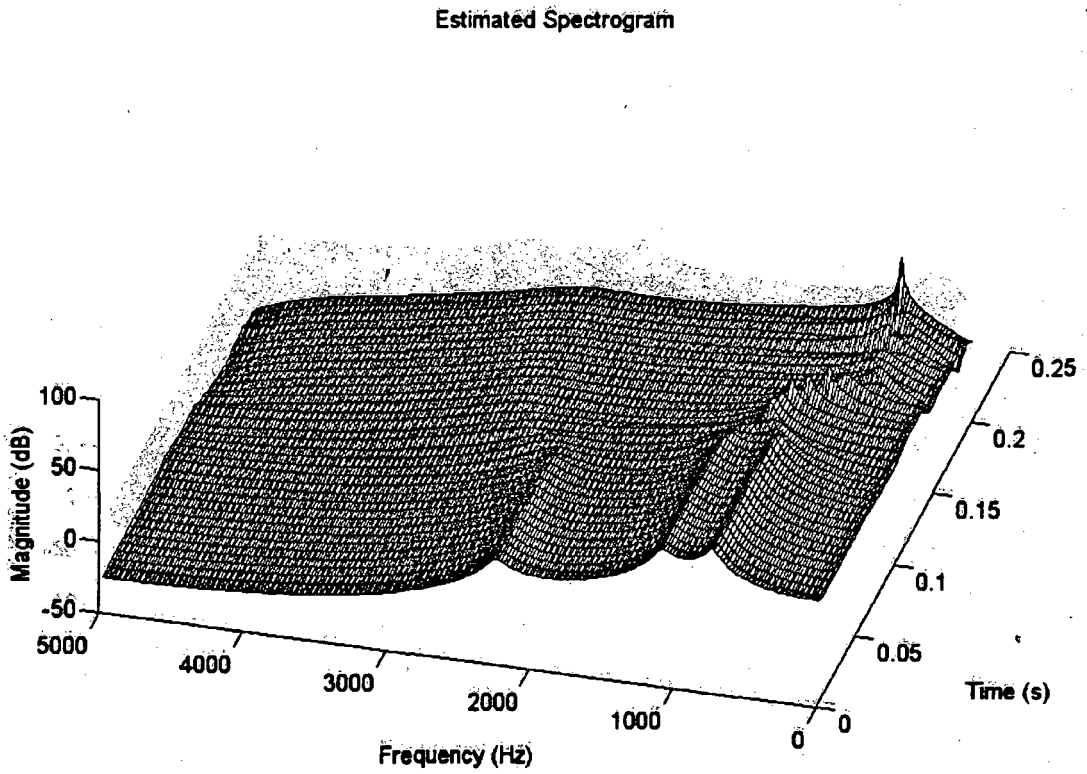


Figure 5.39 Estimated 3D Spectrogram of the synthetic speech signal /a/-/i/ (syn_ai) (WRLS-VFF).

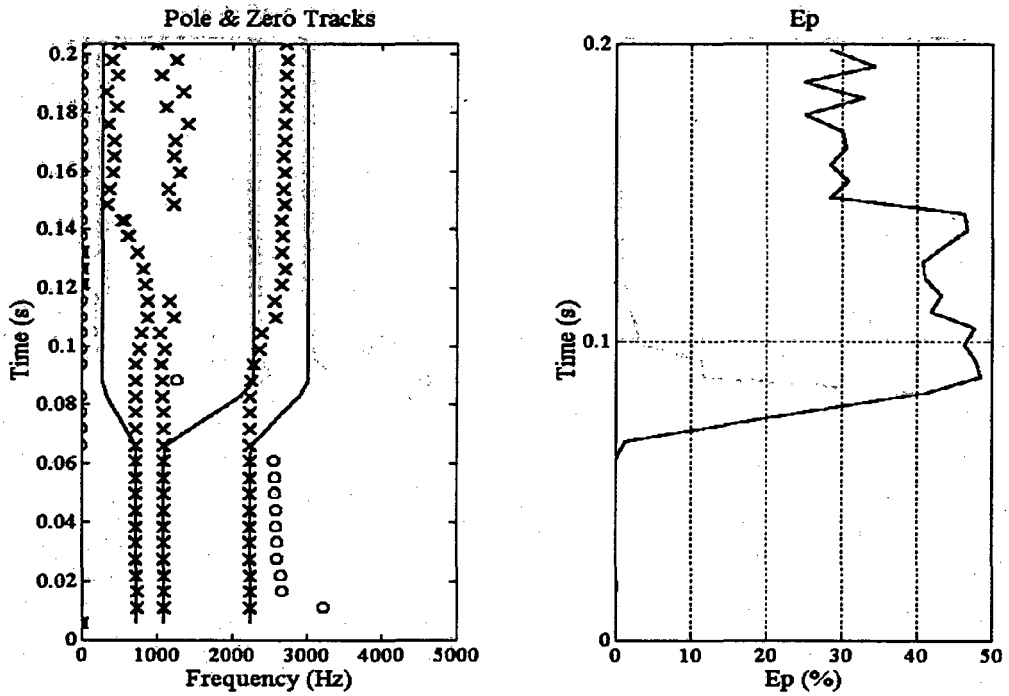


Figure 5.40 Pole-zero tracks of the synthetic speech signal /a/-/i/ (syn_ai) versus the estimated tracks (WRLS-VFF).

Stellenbosch University https://scholar.sun.ac.za

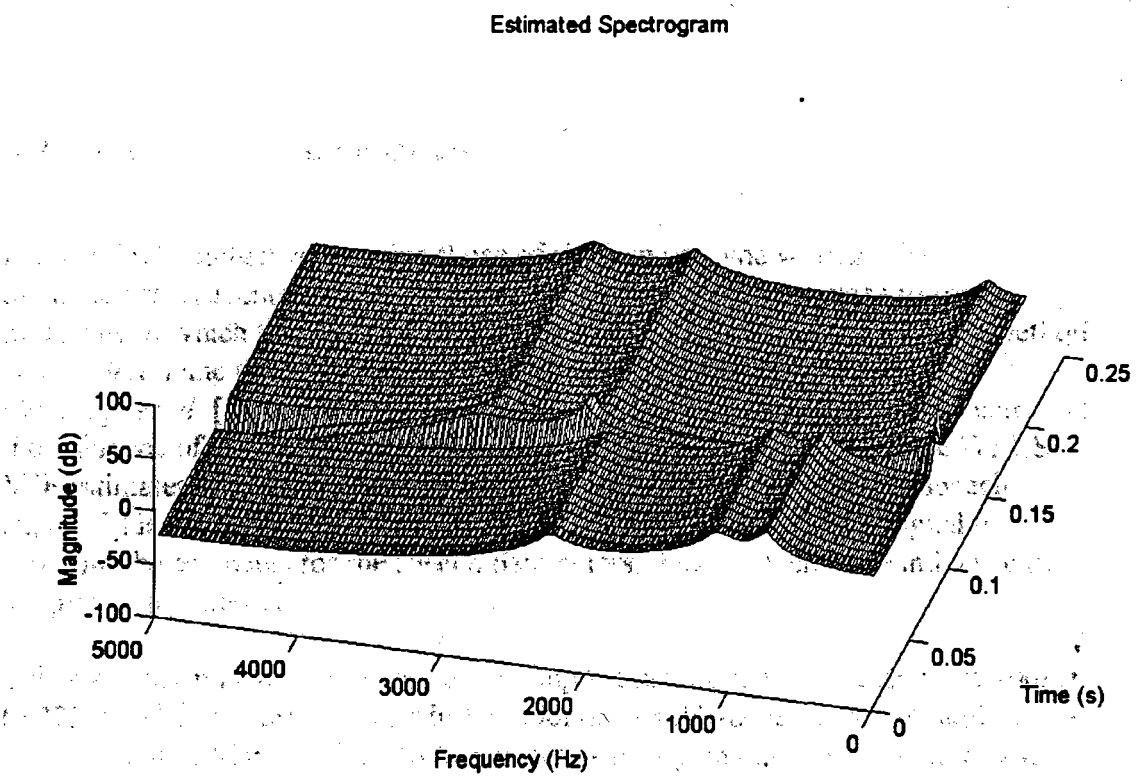


Figure 5.41 Estimated 3D Spectrogram of the synthetic speech signal /a/-/i/ (syn_ai) (FWRLS).

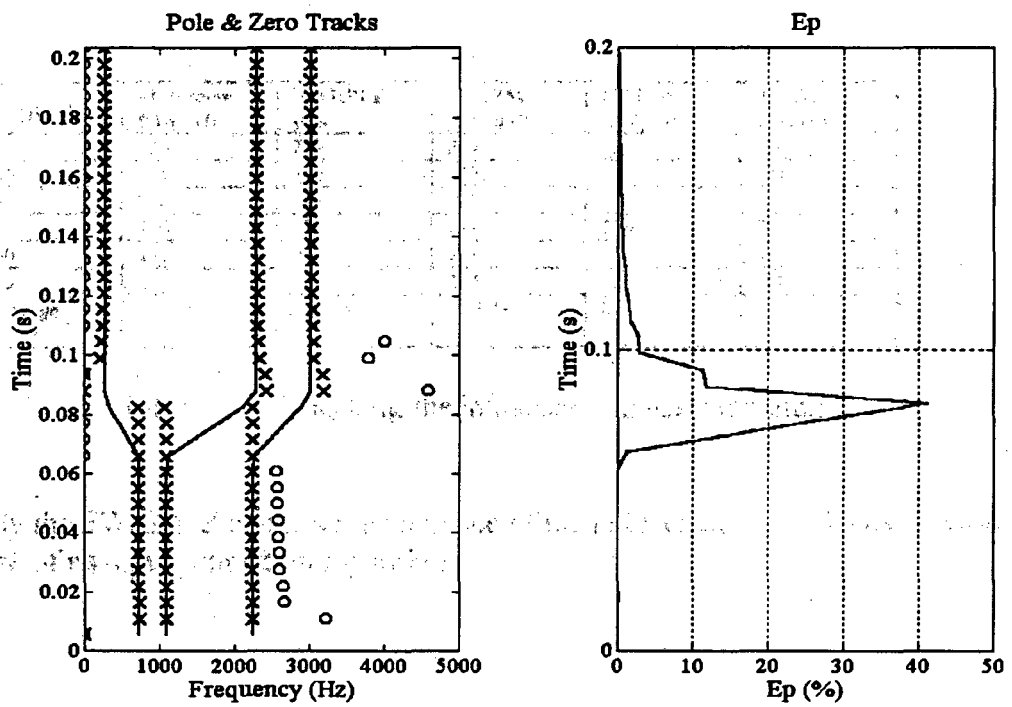


Figure 5.42 Pole-zero tracks of the synthetic speech signal /a/-/i/ (syn_ai) versus the estimated tracks (FWRLS).

Stellenbosch University https://scholar.sun.ac.za

5.3.2. Pitch pulse cancellation

In many high pitched voices, like those of children or some women, the fundamental frequency (pitch pulse frequency) of the speech is close to the frequency at which the first formant occurs. In these cases the normal LPC method cannot determine the frequency of the first formant accurately, as shown by Miyanaga *et al.* [11]. In order to overcome this problem it is necessary to eliminate the influence of the pitch pulses that occur in normal voiced speech. The WRLS-VFF estimates the input signal to the vocal tract by using the available speech signal. This input signal is then used in conjunction with the speech signal to compute the estimates for the speech parameters, thus cancelling the influence of the pitch input excitation.

This was shown in an experiment by varying the pitch pulse frequency from 100Hz to 225Hz when the first formant lies at 250Hz. An AR-order of 6 was used for the LPC method of Marple. The ARMA order was p=6 and q=2 (for the poles and zeros respectively) in the SEARMA and FWRLS methods. The influence of the fundamental frequency on the estimation of the first formant is shown in figure 5.6.

F_0 (Hz)	First formant freq. (Hz)	MARPLE p=6	SEARMA p=6, q=2	FWRLS (AR) p=6	FWRLS (ARMA)
100	250	249	247	250	250
125	250	252	253	250	250
150	250	263	260	250	250
175	250	235	228	250	250
200	250	223	231	250	250
225	250	235	233	250	250
Average Error	-	12.2 %	12.3 %	0 %	0 %

Table 5.6 Cancelling the influence of pulse excitation.

Clearly the FWRLS eliminates the influence of the pitch excitation effectively with the use of its input estimation algorithm.

Stellenbosch University https://scholar.sun.ac.za

5.4. ADDITIONAL APPLICATIONS OF THE WRLS-VFF ALGORITHM

5.4.1. Pitch Detection

Hübing [15] showed that the VFF from the WRLS-VFF can be used as an accurate alternative to residual-based pitch extraction. The interested reader may consult the relevant article.

5.4.2. Formant Tracking

From experiments done on synthetic speech it is clear that the FWRLS is capable of following both formants and anti-formants as well as their bandwidths with greater accuracy than most other sequential methods.

Two spectrograms of real speech, first analysed with the block technique of Marple and then by using the FWRLS are shown in figures 5.43 and 5.44 respectively.

Stellenbosch University https://scholar.sun.ac.za

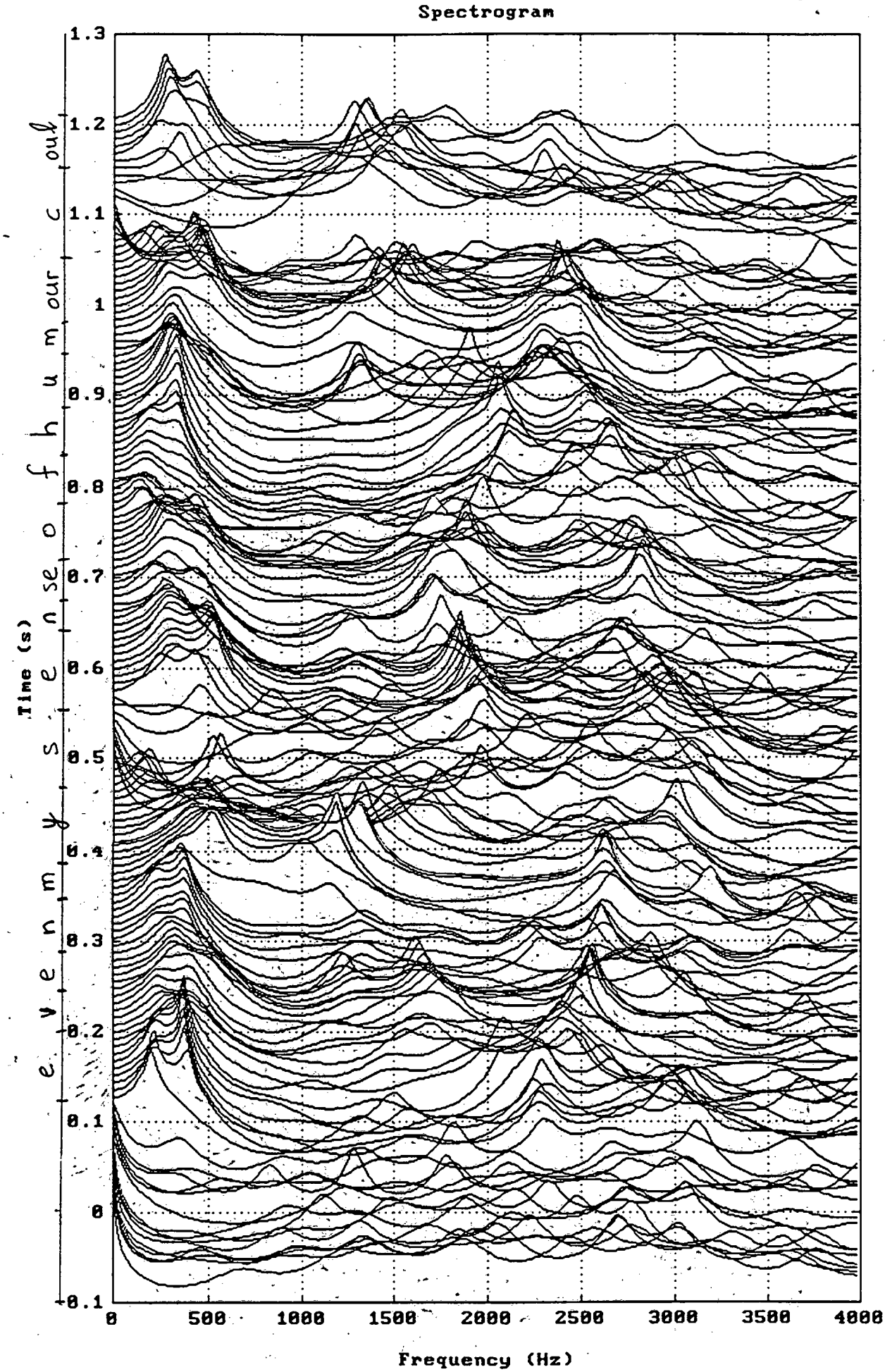
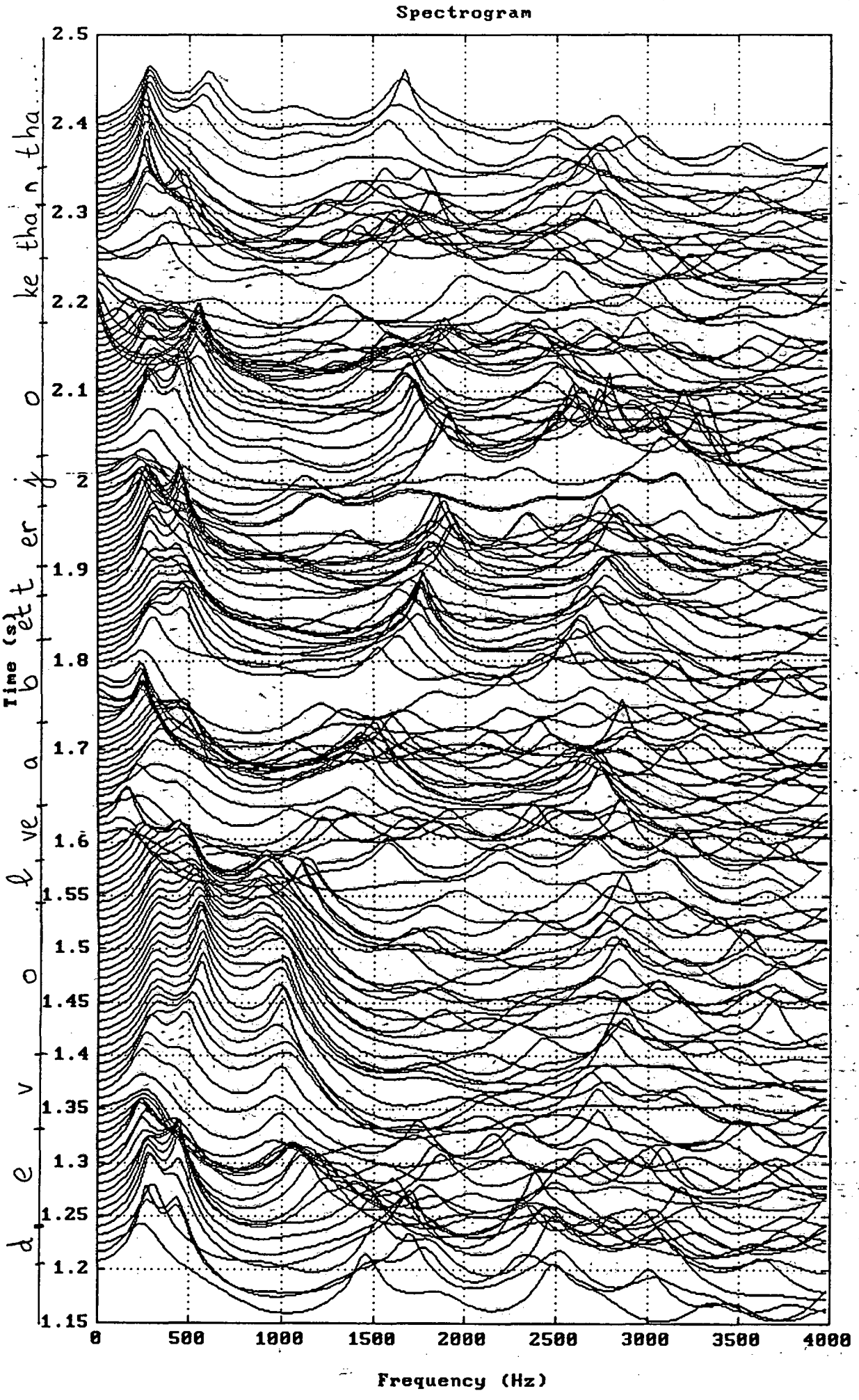


Figure 5.43 Estimated spectrogram for real speech signal (MARPLE technique).

Stellenbosch University https://scholar.sun.ac.za



Stellenbosch University https://scholar.sun.ac.za

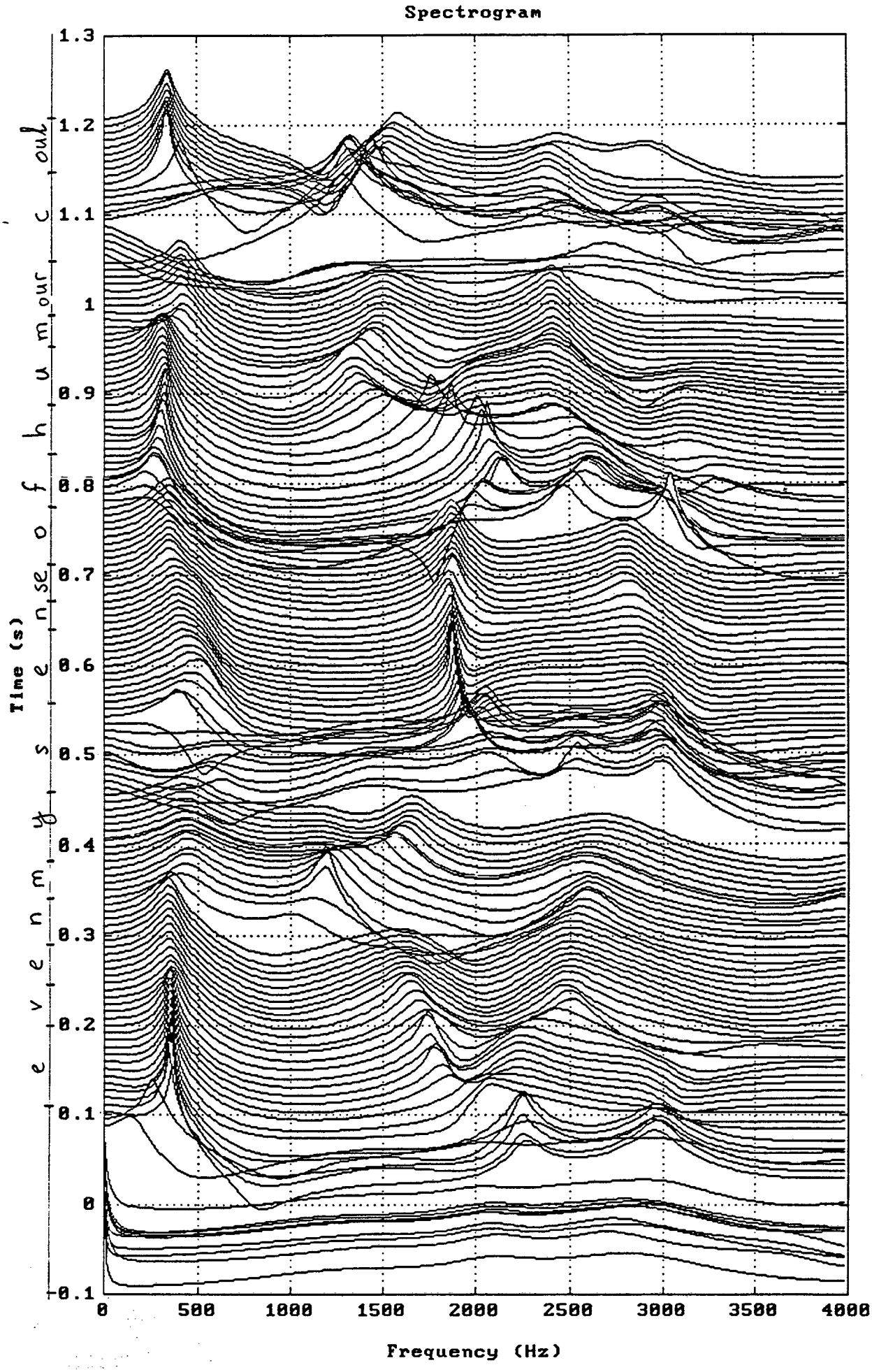
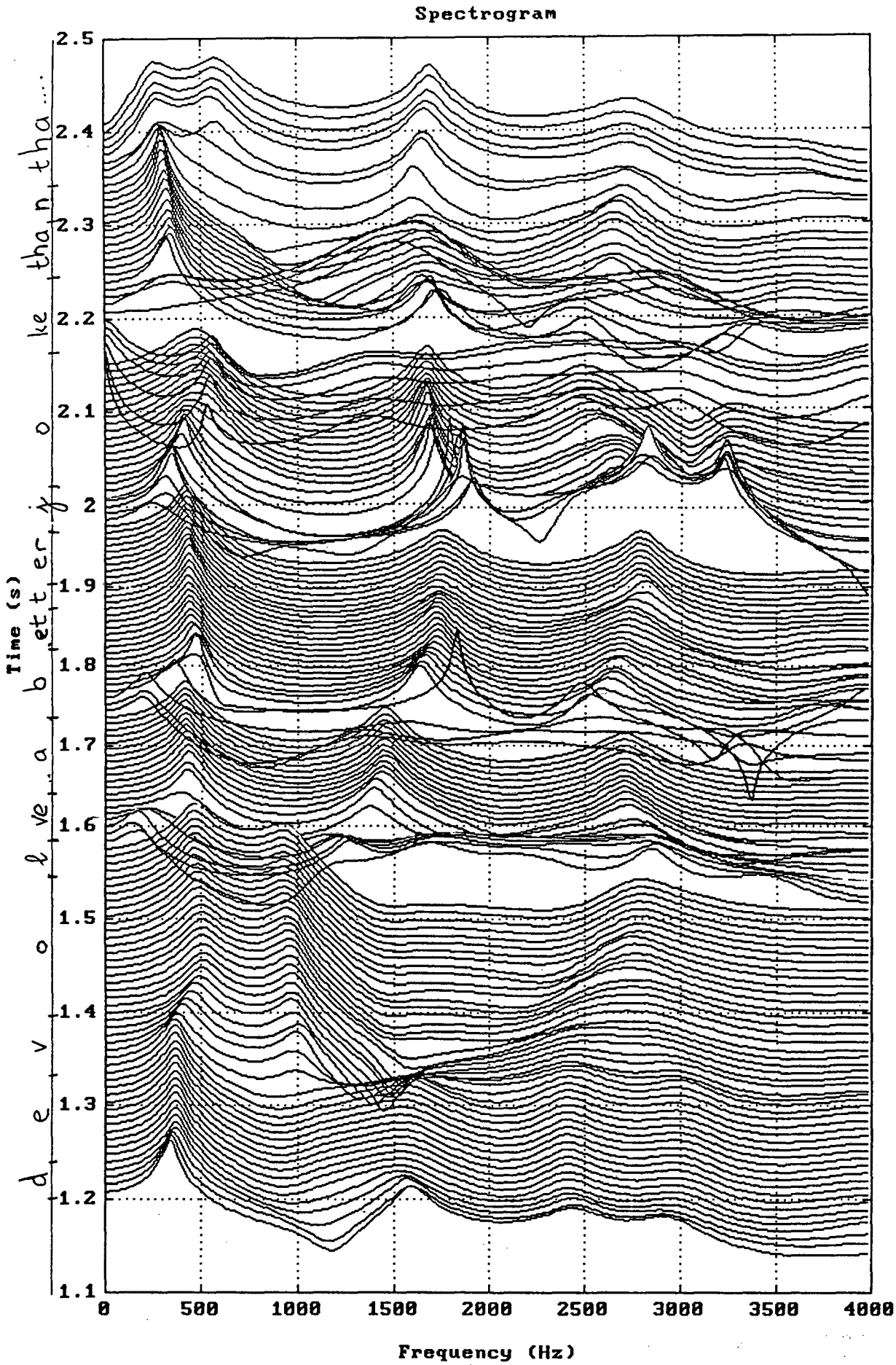


Figure 5.44 Estimated spectrogram for real speech signal (FWRLS technique).

Stellenbosch University https://scholar.sun.ac.za



Stellenbosch University https://scholar.sun.ac.za

5.4.3. Noise Rejection

All previous testing of the WRLS-VFF algorithm was done on noise-free signals. However, the addition of white noise to an AR process results in a degradation of the spectral estimates [51]. It is well known that the resolution of the AR spectral estimator decreases as the signal-to-noise ratio (SNR) decreases [50],[51]. Kay [51] explains that this happens because the power spectral density (PSD) of the signal contaminated with white noise, is already flat, hence any linear prediction filtering will not significantly whiten the PSD any further. One method to reduce the effect of the noise on the AR spectral estimates is to increase the model order. This is because the appropriate model for an AR process in white noise is an ARMA model, and an ARMA model can be approximated by a high order AR process [50]. Another method is to use an ARMA estimation algorithm directly.

5.5. CONCLUSION

Different sequential algorithms were compared by using synthetic speech signals. The proposed method (FWRLS) achieved the best results in:

- The detection of pole and zero bandwidths and frequencies.
- The tracking of pole and zero frequencies.

Another experiment showed that the input estimation algorithm eliminates the influence of pitch pulses on the parameter estimates, as anticipated.

Stellenbosch University https://scholar.sun.ac.za

Chapter 6

Speech Analysis

6.1. INTRODUCTION

ARMA cepstra are used as discriminating features in the recognition process (Chapter 7). In this chapter the formulas for converting the ARMA parameters to cepstra are derived.
Although it was not part of the aim of this project, we derived the necessary formulae to convert an ARMA filter to its Mel-scale representation. It is known that Mel-scale parameters produce better estimates results for vowel recognition [6].

6.2. ANALYSIS METHODS

6.2.1. WRLS-VFF

We tested the WRLS-VFF on isolated word recognition by using the initiation routines as discussed in chapter 4. We initially used a modified form of the WRLS-VFF, namely the FWRLS, which was discussed in chapter 5. In order to compare the FWRLS with a block algorithm it was necessary to store a parameter set at constant intervals, corresponding to the block length used in the frame-based algorithm. By doing this, it sometimes happened that the parameters were stored just after the covariance matrix had been reset by the fractal dimension estimation algorithm. The result was that these parameter sets were erroneous, because the FWRLS did not have enough time to converge to the correct parameter values.

We therefore decided not to use the fractal dimension estimator in this comparison or at least until these wrong parameter sets could be eliminated. Although we refer to the FWRLS during the rest of this chapter, the reader must remember that this is actually the FWRLS without the fractal dimension estimator.

Stellenbosch University https://scholar.sun.ac.za

6.2.2. Marple

The block algorithm of Marple [18] was used as reference to test the effectiveness of the FWRLS on isolated word recognition. It is possible to fine-tune the algorithm by changing various parameters, i.e. the effective block length can vary, the AR order can be changed and blocks can overlap each other to provide a smoother spectrogram.

When we applied the Marple algorithm to speech, we used a Hamming window with a 50% overlap between frames of length 16ms or 32ms. These frame lengths thus produced a parameter vector every 8ms or 16ms. The AR order (and cepstra order) was selected as a constant 12 or 18 to compare it with the FWRLS algorithm. Cepstra coefficients were used as feature vectors in the recognition process.

Stability of AR-systems

One of the implementation problems associated with the Marple algorithm is that a stable filter is not guaranteed. A method of testing the stability of the obtained AR filter had to be found. The stability of a system is determined by the position of the roots of the denominator polynomial (poles) on the transfer function. For an AR-filter, the transfer function is:

$$H(z) = \frac{1}{A_N(z)}$$

$$= \frac{1}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_N z^{-N}}$$

$$= \frac{1}{\sum_{k=0}^{N} a_k z^{-k}}$$

where N=order of the AR system and $a_0, a_1 \ldots a_N$ are the coefficients of the denominator polynomial $A_N(z)$.

This system is causal if, and only if, the roots of $A_N(z)$ lie inside the unit circle. We discuss two different tests for stability of an AR-filter.

Stability Test 1 : Growing Impulse Response

This method computes the impulse response for the given AR filter over a fixed number of samples. If the impulse response grows bigger, over the whole length of

Stellenbosch University https://scholar.sun.ac.za

this interval, then the system is unstable. Whether a given system is classified as stable or unstable is highly influenced by the length of this interval. A better way would be to test directly whether the poles of the system are inside the unit circle.

Stability Test 2 : Schür-Cohn[3]

The Schür-Cohn test for stability [59], [33], [62], [35], computes the reflection coefficients of a polynomial. For a polynomial to be stable (all roots inside the unit circle), the magnitudes of each of the reflection coefficients must be *less* than unity. The reflection coefficients are obtained from the polynomial and its reciprocal by using a recursive formula. The algorithm is described in more detail in Appendix B.

6.3. CEPSTRUM COEFFICIENTS

Traditionally, parameters obtained from linear prediction are transformed to a smoothed representation before they are applied to a speech recognition technique. AR cepstral coefficients, which retain the smoothed spectral behaviour of the speech signal, have been applied successfully to speaker identification and verification by Atal [34]. Juang and Rabiner [36] applied it successfully to isolated word recognition as well as single vowel recognition . Shikano [6] evaluated several LPC spectral matching measures for phonetic unit recognition. He concluded that the cepstrum distance measure fared better than clean AR parameters when tested on phonetic unit recognition.

The above mentioned cepstra coefficients are derived from the AR-LPC parameters. If the speech production process is considered as an ARMA process, these distance measures may not give an accurate representation of the spectral distance of the analysed signals. In the next few sections an ARMA cepstral distance is derived.

6.3.1. Computation

After the AR or ARMA parameters are obtained, the cepstra coefficients are computed. These are then used as feature vectors for speech recognition purposes.

A recursion for transforming AR-LPC parameters to cepstra coefficients was derived by Atal [34]. C.J. van der Merwe [48] generalized this method to include ARMA transfer functions. The deduction is repeated here in more detail:

Let us define the transfer function of an ARMA filter as:

[3]This test can be traced back to Schür (1917) and Cohn (1922).

Stellenbosch University https://scholar.sun.ac.za

$$H(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{\sum_{k=0}^{q} b_k z^{-k}}{\sum_{k=0}^{p} a_k z^{-k}} \qquad a_0 = 1 \qquad 6.1$$

Differentiate both the numerator and denominator with respect to z^{-1}. The zero'th coefficient of each polynomial disappears:

$$\frac{\partial}{\partial z^{-1}} B(z^{-1}) = \sum_{k=1}^{q} k b_k z^{1-k} \qquad 6.2$$

$$\frac{\partial}{\partial z^{-1}} A(z^{-1}) = \sum_{k=1}^{p} k a_k z^{1-k} \qquad 6.3$$

Oppenheim and Schafer [55] define the complex cepstrum as the inverse Z-transform of the complex natural logarithm of the Z-transform of the data sequence. If the transfer function represents a stable filter (i.e. all poles inside the unit circle) then the complex cepstrum function is given by:

$$\sum_{n=0}^{\infty} C_n z^{-n} = ln\left[H(z^{-1})\right] \qquad 6.4$$

Where C_n are the cepstrum coefficients. In order to isolate the cepstrum coefficients we differentiate equation 6.4 on both sides with respect to z^{-1}. Again the zero'th coefficient of the polynomial on the left hand side disappears:

$$\begin{aligned}
\sum_{n=1}^{\infty} nC_n z^{1-n} &= \frac{\partial}{\partial z^{-1}}\left\{ln\left[H(z^{-1})\right]\right\} \\
&= \frac{\partial}{\partial z^{-1}}\left\{ln\left[B(z^{-1})\right] - ln\left[A(z^{-1})\right]\right\} \\
&= \frac{B'(z^{-1})}{B(z^{-1})} - \frac{A'(z^{-1})}{A(z^{-1})} \\
&= \frac{A(z^{-1})B'(z^{-1}) - B(z^{-1})A'(z^{-1})}{A(z^{-1})B(z^{-1})}
\end{aligned}$$

$$\sum_{n=1}^{\infty} nC_n z^{1-n}\left[A(z^{-1})B(z^{-1})\right] = A(z^{-1})B'(z^{-1}) - B(z^{-1})A'(z^{-1})$$

Replace the terms $A(z^{-1})$ and $B(z^{-1})$ with their definitions and use equations 6.3 and 6.2. Keep in mind that $a_i = 0$ for $i > p$ and $b_j = 0$ for $j > q$:

Stellenbosch University https://scholar.sun.ac.za

$$\sum_{n=1}^{\infty} nC_n z^{1-n}\left[\sum_{i=0}^{p} a_i z^{-i}\sum_{j=0}^{q} b_j z^{-j}\right] = \sum_{i=0}^{p} a_i z^{-i}\sum_{j=1}^{q} jb_j z^{1-j} - \sum_{j=0}^{q} b_j z^{-j}\sum_{i=1}^{p} ia_i z^{1-i}$$

$$\sum_{n=1}^{\infty} nC_n \sum_{i=0}^{p}\sum_{j=0}^{q} a_i b_j z^{1-n-(i+j)} = \sum_{i=0}^{p}\sum_{j=1}^{q} ja_i b_j z^{1-(i+j)} - \sum_{i=1}^{p}\sum_{j=0}^{q} ia_i b_j z^{1-(i+j)}$$

$$\sum_{n=1}^{\infty} nC_n \sum_{i=0}^{p}\sum_{j=0}^{q} a_i b_j z^{1-n-(i+j)} = \sum_{i=1}^{p}\sum_{j=1}^{q} (j-i)a_i b_j z^{1-(i+j)} + \sum_{j=1}^{q} ja_0 b_j z^{1-j} - \sum_{i=1}^{p} ia_i b_0 z^{1-i} \qquad 6.5$$

Equate the coefficients of equal powers of z^{-1} to obtain:

$$z^0: \quad C_1 = \frac{a_0 b_1 - a_1 b_0}{a_0 b_0}$$

$$z^{-1}: \quad C_1[a_0 b_1 + a_1 b_0] + 2C_2 a_0 b_0 = 2a_0 b_2 + 2a_2 b_0$$

$$z^{-2}: \quad C_1[a_0 b_2 + a_1 b_1 + a_2 b_0] + 2C_2[a_0 b_1 + a_1 b_0] + 3C_3 a_0 b_0 = a_1 b_2 - a_2 b_1 + 3a_0 b_3 - 3a_3 b_0$$

etc.

We can solve these equations for C_n to obtain the recursive formulation for the ARMA cepstra coefficients:

$$C_n = \frac{1}{na_0 b_0}\left[\sum_{i=0}^{n}(n-2i)a_i b_{n-i} - \sum_{m=1}^{n-1} mC_m \sum_{i=0}^{n-m} a_i b_{n-m-i}\right]$$

$$n \geq 1 \qquad 6.6$$

For an AR process, with $a_0 = 1$ and $b_0 = 1$, equation 6.6 reduces to:

$$C_n = -a_n - \sum_{m=1}^{n-1} \frac{m}{n} C_m a_{n-m} \qquad n \geq 1 \qquad 6.7$$

This is exactly the same result as that obtained by Juang and Rabiner [36] and also by Atal [34].

The zero'th cepstra coefficient is the natural logarithm of the gain of the filter. This can be seen from the following formulation for the filter transfer function:

$$H(z^{-1}) = \frac{G\left(1 + \sum_{k=1}^{q} b_k z^{-k}\right)}{1 + \sum_{k=1}^{p} a_k z^{-k}} \qquad a_0 = 1 \text{ and } b_0 = 1$$

where G is the gain of the filter.

Stellenbosch University https://scholar.sun.ac.za

The complex cepstrum is given by:

$$\sum_{n=0}^{\infty} C_n z^{-n} = ln\left[H\left(z^{-1}\right)\right]$$

$$= ln[G] + ln\left[\frac{1+\sum_{k=1}^{q} b_k z^{-k}}{1+\sum_{k=1}^{p} a_k z^{-k}}\right]$$

The zero'th cepstrum coefficient is then:

$C_0 = ln[G]$

Shikano [6] showed that this cepstra coefficient, representing the power, is important in improving the ability of a recognition system to distinguish vowels from stops.

6.3.2. Order Selection

When cepstra coefficients were obtained from AR or ARMA parameters we selected a shortened cepstra order equal to the sum of the AR and MA parts. This was done because tests on real speech, analysed with the Marple technique with AR order of 12, showed that there was virtually no increase in recognition rate when the cepstra order was selected as 18 instead of 12. We can summarise by saying that, for an AR system a value for the cepstra order equal to the AR order will be sufficient. For an ARMA system a cepstra order as described above is sufficient. This can also be seen from the equations for the ARMA cepstra.

6.3.3. Experiments

Figure 6.45 shows the frequency response of an ARMA system (thick line) against the smoothed spectra (dashed line) from the shortened cepstra coefficients. The AR order is 7 and the MA order is 4. The cepstra order was chosen as 11.

Stellenbosch University https://scholar.sun.ac.za

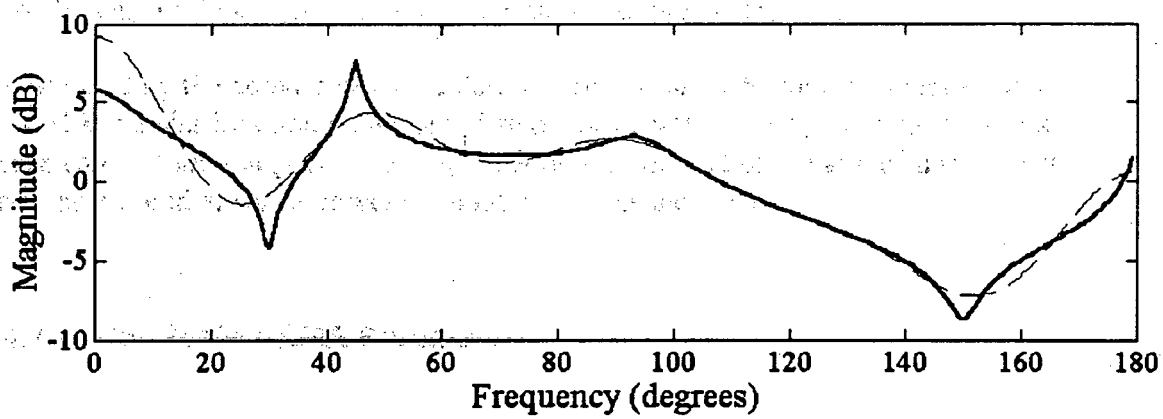


Figure 6.45 True spectrum versus the smoothed spectrum from the shortened cepstra for an ARMA(7,4) model.

6.4. MEL-SCALE REPRESENTATION

The Mel-scale is a perceptually based frequency scale. Equal increments on the mel-scale correspond to equal subjective increments in the frequency [6]. This frequency warping method compresses the linear frequency scale at higher frequencies and expands it at lower frequencies as can be seen in table 6.7.

Linear Frequency (Hz)	Mel-scale frequency (mel)
20	0
160	250
394	500
670	750
1000	1000
1420	1250
1900	1500
2450	1750
3120	2000
4000	2250
5100	2500
6600	2750
9000	3000
14000	3250

Table 6.7 Relationship between Mel- and linear frequency axes [48].

By warping the frequency axes of the cepstra coefficients, Shikano [6] obtained a substantial improvement in vowel recognition. The recognition rates for consonants improved only slightly. Tests conducted by Van der Merwe [48] also showed this improvement in vowel recognition. He transformed the AR-LPC

Stellenbosch University https://scholar.sun.ac.za

coefficients to their Mel-scale representation and then to cepstra.

Motivated by the above results we derived the necessary formulae to represent an ARMA transfer function on a warped frequency scale. The equations for the AR case can be found in [48]. These recursions are not part of the aim of this project, but, as it could enhance vowel recognition, it was included.

6.4.1. Mel-Scale ARMA Parameters

The deduction for converting the ARMA transfer function of equation 2.4 will follow.

$$H(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{\sum_{k=0}^{q} b(k) z^{-k}}{\sum_{k=0}^{p} a(k) z^{-k}}$$

Firstly, apply the unit function $z^{-1} = \frac{\hat{z}^{-1} - \alpha}{1 - \alpha\hat{z}^{-1}}$ by replacing z^{-1} with $\hat{z}^{-1} = \frac{z^{-1} + \alpha}{1 + \alpha z^{-1}}$ in the ARMA transfer function, where α is the warping constant. The new transfer function is:

$$\hat{H}(z^{-1}) = \frac{\hat{B}(z^{-1})}{\hat{A}(z^{-1})} = \frac{\sum_{k=0}^{q} b(k) \left(\frac{z^{-1} + \alpha}{1 + \alpha z^{-1}} \right)^{k}}{\sum_{k=0}^{p} a(k) \left(\frac{z^{-1} + \alpha}{1 + \alpha z^{-1}} \right)^{k}}$$

The new transfer function in rational form as:

$$\hat{H}(z^{-1}) = \frac{\sum_{k=0}^{Q} \hat{b}(k) z^{-k}}{\sum_{k=0}^{P} \hat{a}(k) z^{-k}} \qquad 6.8$$

Note the new orders of the polynomials. Once in this form, the coefficients of the transformed transfer function can be obtained in terms of the original ARMA parameters by equating the coefficients of equal powers of z^{-1}. We shall now rewrite the transfer function in the desired form by multiplying above and below the division line by a factor $(1 + \alpha z^{-1})^{q+p}$:

Stellenbosch University https://scholar.sun.ac.za

$$\hat{H}(z^{-1}) = \frac{\sum_{k=0}^{q} b(k)\left(z^{-1}+\alpha\right)^{k}\left(1+\alpha z^{-1}\right)^{q+p-k}}{\sum_{k=0}^{p} a(k)\left(z^{-1}+\alpha\right)^{k}\left(1+\alpha z^{-1}\right)^{q+p-k}}$$

Apply the binomial formula [58] to expand the products into summations of separate terms.

$$(x+y)^{n} = \sum_{i=0}^{n}\binom{n}{i}x^{n-i}y^{i} \qquad \forall x \neq 0,\ \forall y \neq 0$$

and keep in mind that:

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} = \binom{n}{n-i}$$

The product terms become:

$$\left(z^{-1}+\alpha\right)^{k} = \sum_{i=0}^{k}\binom{k}{i}\alpha^{k-i}z^{-i}$$

and

$$\left(1+\alpha z^{-1}\right)^{q+p-k} = \sum_{i=0}^{q+p-k}\binom{q+p-k}{i}\alpha^{i}z^{-i}$$

The transfer function becomes:

$$\hat{H}(z^{-1}) = \frac{\hat{B}(z^{-1})}{\hat{A}(z^{-1})} = \frac{\sum_{k=0}^{q}\left\{b(k)\cdot\sum_{i=0}^{k}\binom{k}{i}\alpha^{k-i}z^{-i}\cdot\sum_{i=0}^{q+p-k}\binom{q+p-k}{i}\alpha^{i}z^{-i}\right\}}{\sum_{k=0}^{p}\left\{a(k)\cdot\sum_{i=0}^{k}\binom{k}{i}\alpha^{k-i}z^{-i}\cdot\sum_{i=0}^{q+p-k}\binom{q+p-k}{i}\alpha^{i}z^{-i}\right\}}$$

Combine the summations and use equation 6.8:

$$\sum_{k=0}^{Q}\hat{b}(k)z^{-k} = \sum_{k=0}^{q}\left\{b(k)\cdot\sum_{i=0}^{k}\binom{k}{i}\alpha^{k-i}z^{-i}\cdot\sum_{i=0}^{q+p-k}\binom{q+p-k}{i}\alpha^{i}z^{-i}\right\} \qquad 6.9$$

$$= \sum_{k=0}^{q}b(k)\sum_{i=0}^{k}\sum_{j=0}^{q+p-k}\binom{k}{i}\binom{q+p-k}{j}\alpha^{k-i+j}z^{-(i+j)}$$

Stellenbosch University https://scholar.sun.ac.za

$$\sum_{k=0}^{P}\hat{a}(k)z^{-k}=\sum_{k=0}^{p}\left\{a(k)\sum_{i=0}^{k}\binom{k}{i}\alpha^{k-i}z^{-i}\sum_{i=0}^{q+p-k}\binom{q+p-k}{i}\alpha^{i}z^{-i}\right\}$$
$$=\sum_{k=0}^{p}a(k)\sum_{i=0}^{k}\sum_{j=0}^{q+p-k}\binom{k}{i}\binom{q+p-k}{j}\alpha^{k-i+j}z^{-(i+j)} \qquad 6.10$$

Notice that $Q=q+p$ and $P=q+p$ because $i+j\le q+p$. The new transfer function thus has an equal number of zeros (Q) and poles (P), which is the sum of that of the old transfer function. By comparing the coefficients of equal powers in equations 6.9 and 6.14 we have:

$$\hat{b}(k)=\sum_{i=0}^{k}\sum_{n=i}^{q+p-k+i}b(n)\binom{n}{i}\binom{q+p-n}{k-i}\alpha^{n+k-2i} \qquad 6.11$$

for $k=0\ldots q+p$ and with $n=q$ if $n>q$.

$$\hat{a}(k)=\sum_{i=0}^{k}\sum_{n=i}^{q+p-k+i}a(n)\binom{n}{i}\binom{q+p-n}{k-i}\alpha^{n+k-2i} \qquad 6.12$$

for $k=0\ldots q+p$ and with $n=p$ if $n>p$.

Equations 6.11 and 6.12 give the relationship between the ARMA parameters and the Mel-scale coefficients. The warping constant (α) has a value between -1 and +1, but not zero. The transfer function for the Mel-scale representation is given by 6.8.

Notice that computational requirements can be reduced considerably by exploiting the similarity between the two equations.

6.4.2. Mel-Scale AR Parameters

The formulae to represent an AR transfer function on a warped frequency scale can be obtained from 6.11 and 6.12 as follows:
Set $q=0$ and,

$$b(n)=\begin{cases}1 & n=0\\ 0 & n=1,2\ \ldots\end{cases}$$

and

$a(n)=0$

$n=p+1,\ p+2,\ \ldots$

then we have:

$$\hat{b}(k)=\binom{p}{k}\alpha^{k} \qquad 6.13$$

Stellenbosch University https://scholar.sun.ac.za

$$\hat{a}(k) = \sum_{i=0}^{k} \sum_{n=i}^{p-k+i} a(n) \binom{n}{i} \binom{p-n}{k-i} \alpha^{n+k-2i} \qquad 6.14$$

These relations correspond to those found in [48].

6.4.3. Experiments

From 6.11 and 6.12 we see that the new AR and ARMA orders in the warped frequency axis are equal and have a value equal to the sum of the unwrapped AR and MA orders. This means that computation time, for computing the cepstra, is increased enormously. We therefore did not test the enhancement that a Mel-scale representation could bring about in the recognition results which are discussed later. The Mel-scale spectrum of a signal is shown in figure 6.46 where α is increased from 0.2 to 0.8.

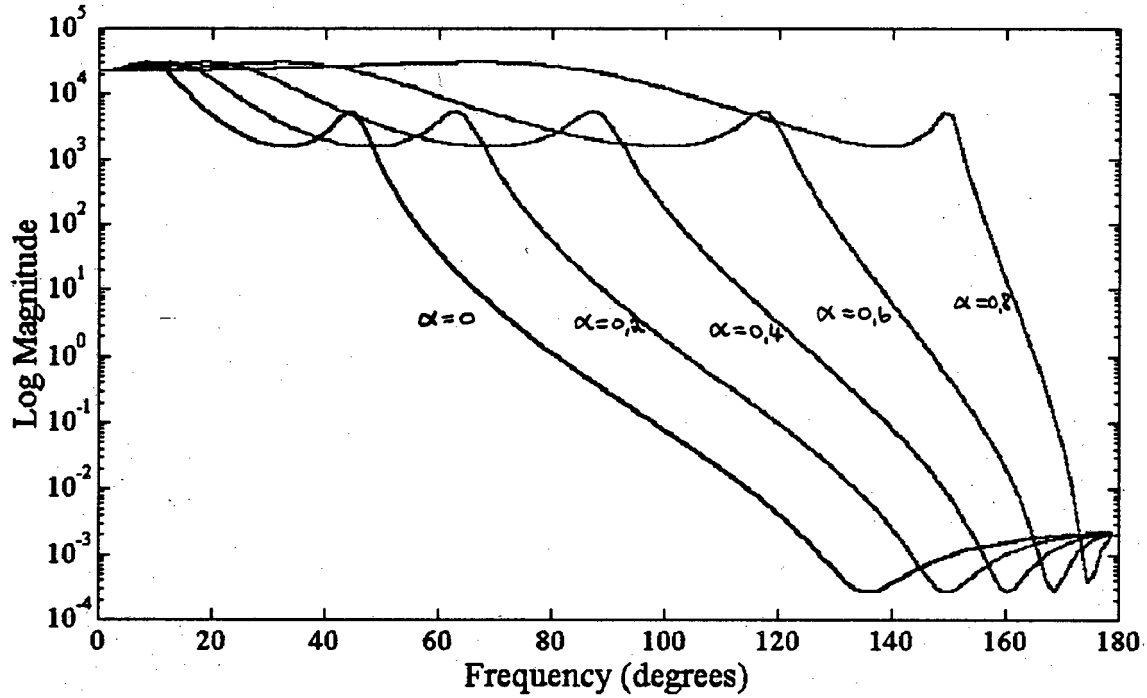


Figure 6.46 Melscale spectrum for an ARMA(4,2) model when α is increased from 0.2-0.8.

Stellenbosch University https://scholar.sun.ac.za

6.5. CONCLUSION

In this chapter we derived the ARMA coefficients and the cepstra coefficients. We also derived the relationship between ARMA parameters and their Mel-scale representation. Unfortunately we could not test the increase in vowel recognition rate due to an enormous increase in computational time for calculating the ARMA cepstra features.

Stellenbosch University https://scholar.sun.ac.za

Chapter 7

Isolated Word Recognition

7.1. INTRODUCTION

In this chapter we shall compare an ARMA sequential technique with the popularly used block algorithm by Marple [18]. The comparison is done by testing both algorithms on isolated word recognition. ARMA cepstra are used as discriminating features in the recognition process.

7.2. PREPROCESSING THE SPEECH

- Before extracting the parameters from a speech file, the speech was normalised so that the magnitude was bounded by -0.5 and 0.5.

- The speech was pre emphasized before extracting the parameters in order to suppress the fundamental frequency and to reduce the roll-off effect of the glottal source in voiced speech. This usually helps with extracting higher frequency formants of low amplitude. The level of pre-emphasis can be controlled by a constant α, which determines the cut-off frequency of the single-zero filter [49]. The difference equation is:

  $$y_k = s_k - \alpha s_{k-1}$$

  In this equation y_k is the output of the pre-emphasis filter and s_k the input speech signal at time instant k. In the analysis of speech we used a pre-emphasis constant of $\alpha = 0.98$.

7.3. THE TELAZ DATABASE

The telaz speech database, from the University of Stellenbosch, was used in all the experiments conducted in this chapter, unless stated otherwise. This database contains separate words sampled at 8kHz specifically for use in testing isolated

Stellenbosch University https://scholar.sun.ac.za

word recognition systems. Each database file consists of approximately forty one different words read out in random order by different speakers. A list of valid words can be found in appendix F. Transcription files, describing where each word starts and ends, were also used for easier testing.

7.4. SPEECH RECOGNITION

Hidden Markov models (HMM's) were used to set up a model for each of the available 41 words. The design of HMM's is a subject in its own right and will not be discussed any further. The interested reader may consult [37], [38] or [39].

7.4.1. HMM Training

The training data was obtained from 104 files read by 40 different speakers. This represents a total of about 4000 words.

Before training the Hidden Markov Models (HMM), the data (cepstra features in this case) had to be normalised. This was done by normalising each dimension of the *training set* by the standard deviation of the specific dimension.

A simple, 10 state, left to right model (HMM) was used for each word.

7.4.2. HMM Testing

A total of 24 files read by 12 different speakers was used for testing the accuracy of the HMM's for each of the 41 words in the Telaz database. The test-speakers were different from those used in training the HMM's.

7.5. EXPERIMENTS AND RESULTS

In order to compare the frame based Marple algorithm with the sequential FWRLS algorithm we did the following:

Exactly the same training and testing data were used for both algorithms. The Marple algorithm was done for two frame lengths namely 16ms and 32 ms with a 50% overlap between frames. These results were compared to those of the FWRLS when a set of parameter values was stored every 8ms and 16ms respectively. The AR- and cepstra orders in the Marple algorithm were kept constant on 12 and 18 poles respectively. In the FWRLS a constant AR order of

Stellenbosch University https://scholar.sun.ac.za

12 and MA order of 6 were used with a truncated cepstra order of 18.

Results of these tests are shown in table 8.8.

Method	Frame length or update time (ms)	AR order	MA order	Truncated Cepstra Order	Train Speakers (%)	Test Speakers (%)
Marple	16	12	-	12	94.7	92.3
Marple	32	12	-	12	96.5	95.1
Marple	16	18	-	18	95.9	93.8
Marple	32	18	-	18	97.0	95.9
FWRLS	4	12	6	18	95.3	83.3
FWRLS	8	12	6	18	94.4	83.1
FWRLS	16	12	6	18	92.9	82.8

Table 8.8 Results of isolated word recognition tests.

It would have been more sensible to compare the AR block technique with a certain constant order, against the sequential algorithm with the same AR order and an MA order of zero. We decided against this for practical reasons associated with the computational load of the FWRLS algorithm. We can further argue that, if the ARMA sequential algorithm has difficulty in obtaining the same recognition rate as that of the Marple algorithm, then the AR sequential algorithm will have a slightly lower rate, in which case the latter will be a time consuming, power exercise to show what we already know.

From table 8.8 we draw the following conclusions:

- Both the parameter estimation techniques (the block and the sequential) obtained high recognition rates for the training data set, indicating that the possibility is there for good recognition rates on the testing data, if the training data is enough.

- In all the tests with the Marple block technique the difference between the training and testing results is very small (1.1%-2.4%). This indicates that the novel data (training set) was sufficient for the Marple technique.

- The same cannot be said about the difference between the training and testing results in the case of the FWRLS algorithm. The difference is as high as ~10%, which indicates that there is too little training data. We can only speculate on the reason for this. One explanation might be that more speakers are needed for higher variances of the normal distributions in the HMM's. The parameters' tracks obtained by the Marple algorithm are much more random than those of the FWRLS algorithm which will automatically result in wider normal distributions.

- We see that when the model order of the block method was increased, the recognition rate went up. We postulate that this could be as a result of better estimation of zeros in the signal, because an MA process can be approximated by a high order AR process [50], [51].

Stellenbosch University https://scholar.sun.ac.za

- A longer window length for the block technique increases the results dramatically. The reason for the increase is that more accurate autocorrelation estimates can be obtained from the longer window. If the window is too long the recognition rate will drop again, because we assume the speech signal to be stationary in the analysis window.

- A shorter interval between the updating of the parameters, in the case of the FWRLS method, increases the recognition rate. This is a direct result of more data being made available for training and testing during the recognition process.

When comparing the two algorithms with each other we notice the following:

The recognition rates for the training data of the FWRLS method is, at its best, 0.6% better than that of the worst AR block method when using a window length of 16ms. The FWRLS method did about 1.7% worse than the best results obtained by the Marple technique when using the 32ms window. We hope to close this gap in future research, by using a larger set of training data for the FWRLS algorithm. This may also push up the recognition rate for the test data.

7.6. CONCLUSION

Our current frame based LPC technique (Marple's algorithm) for analysing speech was compared to the proposed sequential technique (FWRLS without the fractal estimator). This was done by applying both techniques to isolated word recognition. AR/ARMA cepstra were used as features for the HMM training/testing. We can summarise the results by stating that, with the available database, it is impossible to say which of the two algorithms fared better. It seems as if the sequential technique only needs more training data to obtain the same or better recognition rates than the block algorithm.

Stellenbosch University https://scholar.sun.ac.za

Chapter 8

Conclusion

8.1. SUMMARY

An adaptive method for the estimation of speech parameters was implemented.

- An RLS-based algorithm in a system identification situation was used. An effective input estimation algorithm [14] laid the foundation for eliminating the effect of pitch pulses on the estimated parameters. Although the sequential algorithm by Miyanaga *et al* [11] also alleviated the influence of the excitation source on the parameters, their method used two adaptive algorithms to achieve this. The result was a much more computationally complex algorithm.

- Non-stationary signals can be followed faster and with greater accuracy than with previously available techniques. This is achieved by employing a variable forgetting factor which will automatically increase or reduce the effective memory of the algorithm. The SEARMA method of Morikawa and Fujisaki [7] is applicable to stationary signals only. The conventional RLS method with a constant (but smaller than unity) FF can be applied to non-stationary systems, but it can introduce unnecessary misadjustment for stationary signals.

- A fractal dimension estimator will find the non-linear jumps associated with voiced to unvoiced transitions. This dramatically increases tracking in these areas. None of the sequential methods tested against the FWRLS were capable of following these rapid vowel-consonant transitions.

- The Mel-scale representation of an ARMA transfer function was derived. It was not tested on isolated word recognition due to the computation time required for calculating the ARMA cepstra.

Stellenbosch University https://scholar.sun.ac.za

When compared to the currently popular block technique of Marple we found the following:

- It seems as if the proposed algorithm needs a larger database than that required by the Marple algorithm. This is concluded from the ~10% difference between testing and training results of the sequential algorithm as opposed to the 1.5% difference of the block method.
- The training results of the sequential algorithm compared favourably, even when using the too small database (from 0.6% better to 1.7% worse). This creates expectations of even better results when the training data is increased to narrow the 10% gap between training and testing results.

The FWRLS method can, without any modifications, be applied to accurate formant/anti-formant tracking as showed in chapter 5. Another immediate application is when it is used as an accurate alternative to residual-based pitch extraction [15].

8.2. FUTURE RESEARCH

Pitch synchronous processing

Hubing and Yoo [15] showed that parameter trajectories or the variable forgetting factor produced by the WRLS-VFF can be used to extract voicing, pitch and pitch pulse location information. By selecting the parameters at the point of convergence, just prior to the next pitch pulse, very accurate LPC estimates can be obtained which would result in even better recognition rates.

Faster algorithms

The obvious disadvantage of the WRLS-VFF algorithm is its high computational load. The recursive least squares ladder algorithm (LSL) developed by Lee *et al.* [34] is a fast, robust algorithm with excellent convergence properties. The LSL algorithm was however developed for the RLS method with a constant forgetting factor. There is definitely scope for research on a LSL based algorithm using a variable forgetting factor.

The computation of the update for the covariance matrix takes up the largest slice of the computational time. Gavrishchuk and Starkov [33] succeeded in establishing a series of new relations which allow one to reduce the number of operations in calculating the covariance matrix of the estimation errors. The proposed method of solution is based on the analytic properties of the Ricatti equation.

Stellenbosch University https://scholar.sun.ac.za

Varying the model order

If we could vary the model order, to be more representative of the true order of the process being estimated, more accurate estimates of the parameters would be obtained [7]. Changing the ARMA order will result in feature vectors of different lengths, which is difficult to realize in a HMM recognition system. However, by using the cepstra representation of the parameters, it is possible to obtain feature vectors of the same length regardless of the original ARMA order. The only consequence of using a cepstra order that is higher than the original ARMA order, is that in such a case the cepstra vector would be a more accurate representation of the estimated spectrum!

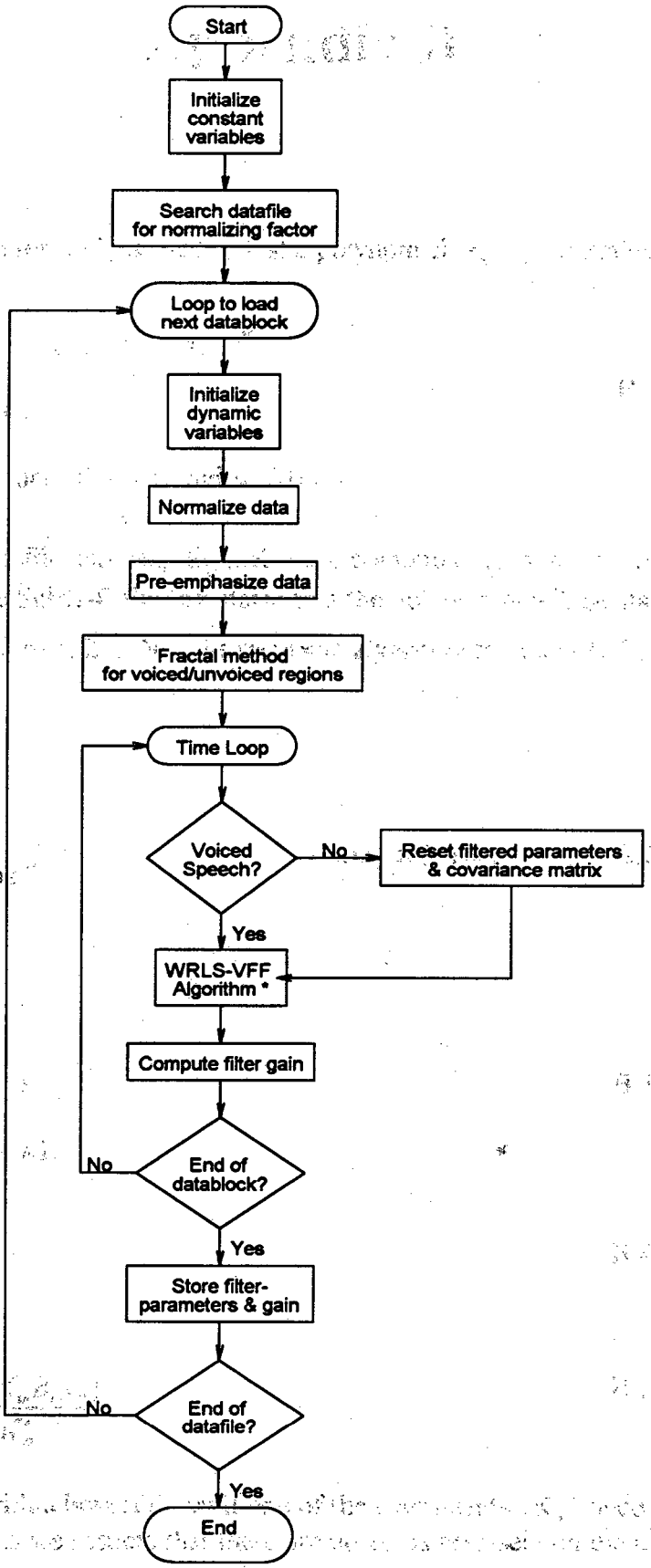
Stellenbosch University https://scholar.sun.ac.za

Appendix A.

Flowchart 1 A Block diagram of the FWRLS method is showed on the following page.

* Note that the blockdiagram of the WRLS-VFF can be found in §3.4 of the text.

Stellenbosch University https://scholar.sun.ac.za



Stellenbosch University https://scholar.sun.ac.za

Appendix B.

The Schür-Cohn test [59] is used to test a polynomial $A_N(z)$ for stability:

$$A_N(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_N z^{-N} = \sum_{k=0}^{N} a_N(k) z^{-k} \qquad \text{B.1}$$

where N is the order of $A_N(z)$ and $a_N(0) = 1$.

The reflection coefficients $\{K_1, K_2, \ldots K_N\}$ are computed from $A_N(z)$ and its reverse polynomial. The Schür-Cohn test states that the polynomial will be stable, if and only if, $|K_m| < 1$ for all m=1, 2, ...N. The recursive algorithm to compute K_m for m=1, 2, ...N, follows:

$$A(z) = A_m(z) = \sum_{k=0}^{m} a_m(k) z^{-k} \quad \text{with } a_k(0) = 1 \qquad \text{B.2}$$

$$B(z) = B_m(z) = z^{-m} A_m(z^{-1}) = \sum_{k=0}^{m} a_m(m-k) z^{-k} \qquad \text{B.3}$$

$$K_m = a_m(m) \qquad \text{B.4}$$

$$A(z) = A_{m-1}(z) = \frac{A_m(z) - K_m B_m(z)}{1 - K_m^2} \qquad \text{B.5}$$

Note: The algorithm breaks down if one of the coefficients $\{K_m\}$ becomes exactly equal to ±1. Thus we assume that there are no zeros precisely on the unit circle.

Stellenbosch University https://scholar.sun.ac.za

Appendix C.

The Principle of Orthogonality [56]

Let us define the estimation error in a certain adaptive system as:

$$e_k = y_k - \hat{y}_k \qquad k = i_1 \ldots i_2 \qquad \text{C.1}$$

where

$$\hat{y}_k = -\sum_{i=1}^{p} a_k(i) y_{k-i} + \sum_{i=1}^{q} b_k(i) u_{k-i} \qquad k = i_1 \ldots i_2 \qquad \text{C.2}$$

$$\hat{y}_k = -\sum_{i=1}^{p+q} \theta_k(i) \phi_k(i) \qquad k = i_1 \ldots i_2 \qquad \text{C.3}$$

Where, p is the poles of the AR system and q the zeros in the MA system and:

$$\theta_k^t = \begin{bmatrix} a_k(1) & a_k(2) & \cdots & a_k(p) & b_k(1) & b_k(2) & \cdots & b_k(q) \end{bmatrix} \qquad \text{C.4}$$

$$\phi_k^t = \begin{bmatrix} -y_{k-1} & -y_{k-2} & \cdots & -y_{k-p} & u_{k-1} & u_{k-2} & \cdots & u_{k-q} \end{bmatrix} \qquad \text{C.5}$$

Consider the following cost function:

$$E(\theta, i_{1,2}) = \sum_{k=i_1}^{i_2} |e_k|^2 \qquad k = i_1 \ldots i_2 \qquad \text{C.6}$$

To optimize the adaptive filter, we minimize $E(\theta, i_{1,2})$ by differentiating with respect to θ. Let $\theta_k(i) = \alpha_k(i) + j\beta_k(i)$ then we can substitute this relationship as well as equation C.2 into the cost function and differentiate with respect to θ:

$$\nabla E = \frac{\partial E}{\partial \alpha_k} + j \frac{\partial E}{\partial \beta_k} \qquad i = 1 \ldots p+q \qquad \text{C.7}$$

Stellenbosch University https://scholar.sun.ac.za

$$\frac{\partial E}{\partial \alpha_k} = \frac{\partial}{\partial \alpha_k}\left(\sum_{k=i_1}^{i_2}|e_k|^2\right)$$

$$= \frac{\partial}{\partial \alpha_k}\left(\sum_{k=i_1}^{i_2} e_k e_k^*\right)$$

$$= \frac{\partial}{\partial \alpha_k}\left(\sum_{k=i_1}^{i_2}\left[y_k - \sum_{i=1}^{p+q}\left(\alpha_k(i) + j\beta_k(i)\right)\phi_k(i)\right]\left[y_k - \sum_{i=1}^{p+q}\left(\alpha_k(i) + j\beta_k(i)\right)\phi_k(i)\right]^*\right)$$

$$= \sum_{k=i_1}^{i_2}\left[-e_k\phi_k^*(i) - \phi_k(i)e_k^*\right]$$

C.8

$$\frac{\partial E}{\partial \beta_k} = \sum_{k=i_1}^{i_2}\left[je_k\phi_k^*(i) - j\phi_k(i)e_k^*\right] \qquad \text{C.9}$$

$$\nabla E = \sum_{k=i_1}^{i_2}\left[-e_k\phi_k^*(i) - \phi_k(i)e_k^*\right] + j\sum_{k=i_1}^{i_2}\left[je_k\phi_k^*(i) - j\phi_k(i)e_k^*\right] \qquad \text{C.10}$$

Set $\nabla E = 0$, then we obtain the result:

$$\sum_{k=i_1}^{i_2}\left(e_k^*\right)\phi_k(i) = 0 \qquad i = 1\ldots p+q \qquad \text{C.11}$$

We interpret equation C.11 as stating that, *for the adaptive filter to achieve its minimum least squares condition, each time series represented by the elements of the input vector ϕ_k, must be orthogonal to the estimation error e_k* (see Haykin [56]). By making small changes to equation C.11, it can be used as a basis for deducing all the least squares methods.

Stellenbosch University https://scholar.sun.ac.za

Appendix D.

Complexity analysis of the WRLS-VFF algorithm:

WRLS-VFF equations	Number of multiplications & additions
$\alpha_k = y_k - \phi_k^t \hat{\theta}_{k-1}^*$	$2(p+q)+1$
$K_k = \dfrac{P_{k-1}\phi_k}{\left(\lambda + \phi_k^H P_{k-1}\phi_k\right)}$	$2(p+q)^2+3(p+q)+1$
$\lambda_k = 1-\alpha_k^2\left(1-\phi_k^t K_k\right)^2 \Big/ E_0(\theta)$	$2(p+q)+5$
If $\lambda_k < \lambda_0$ (Pulse input)	
$u_k^w = 0$	
$u_k^p = y_k - \phi_k^t \hat{\theta}_{k-1}^*$	
If $\lambda_k \geq \lambda_0$ (White noise input)	
$u_k^w = \alpha_k\left(1-\phi_k^t K_k\right)$	
$u_k^p = 0$	
$\hat{\theta}_k = \hat{\theta}_{k-1} + K_k\left(y_k^* - \phi_k^H \hat{\theta}_{k-1} - \hat{u}_k^{*p}\right)$	$2(p+q)+1$
$P_k = \lambda^{-1}\left[P_{k-1} - K_k \phi_k^H P_{k-1}\right]$	$3(p+q)^2$

This bring us to a total of $5(p+q)^2+9(p+q)+8$ multiplications and additions per sample.

Stellenbosch University https://scholar.sun.ac.za

Appendix E.

This appendix contains graphs of zero tracks and graphs of the spectral distance error for the synthetic signals syn_mi and syn_ai. The data in the plots were obtained in the tests on these two signals in chapter 5.

Stellenbosch University https://scholar.sun.ac.za

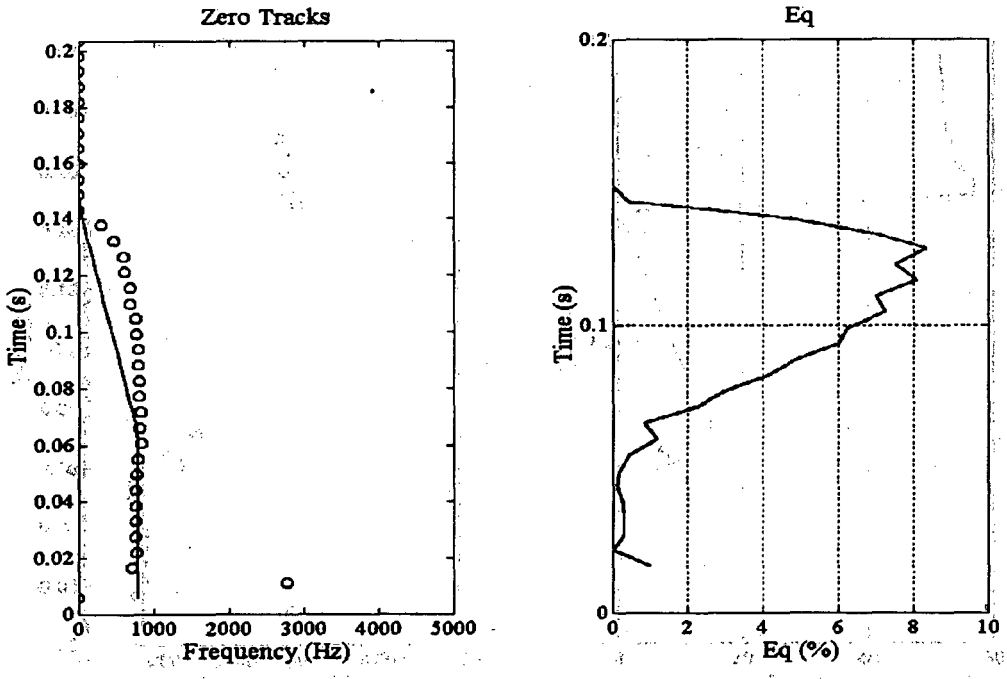


Figure E.1 Zero tracks of the synthetic speech signal /m/-/i/ (syn_mi) versus the estimated tracks (SEARMA).

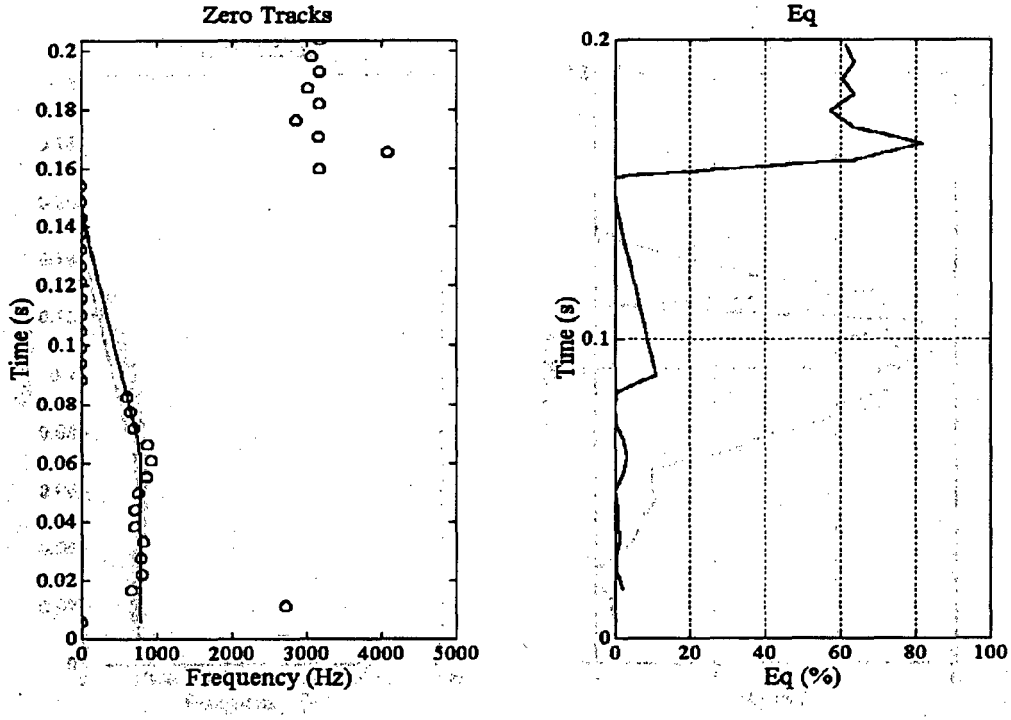


Figure E.2 Zero tracks of the synthetic speech signal /m/-/i/ (syn_mi) versus the estimated tracks (CWRLS).

Stellenbosch University https://scholar.sun.ac.za

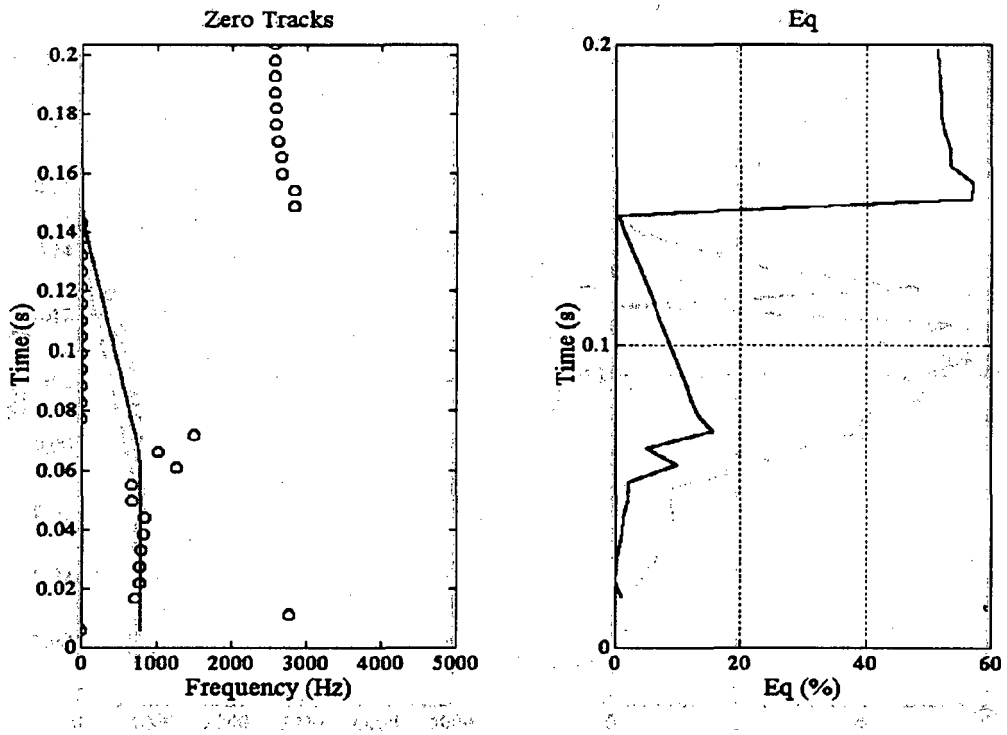


Figure E.3 Zero tracks of the synthetic speech signal /m/-/i/ (syn_mi) versus the estimated tracks (MWRLS).

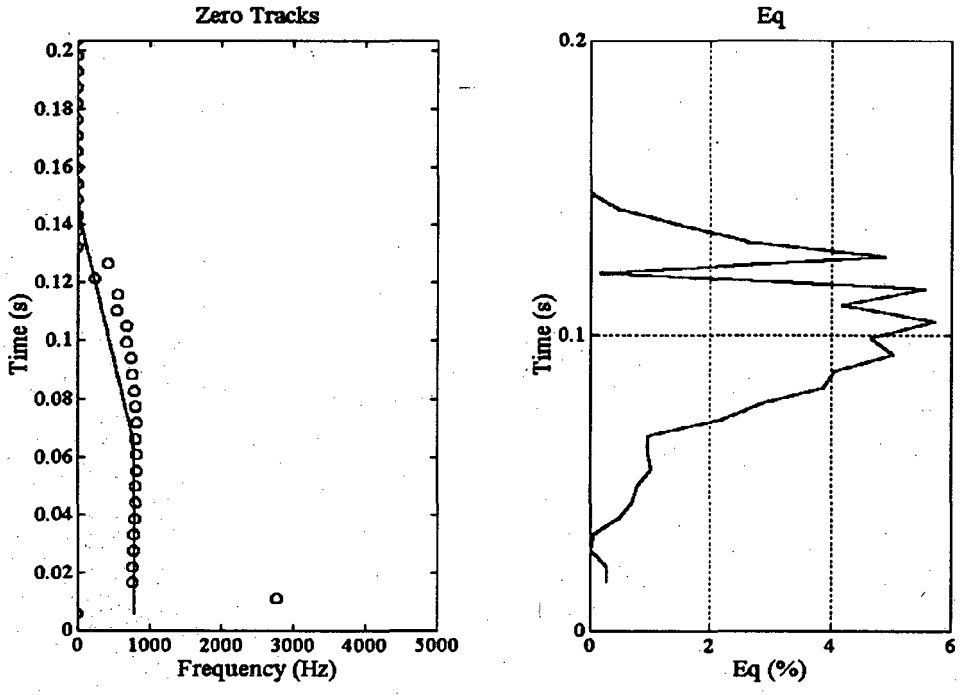


Figure E.4 Zero tracks of the synthetic speech signal /m/-/i/ (syn_mi) versus the estimated tracks (WRLS-VFF).

Stellenbosch University https://scholar.sun.ac.za

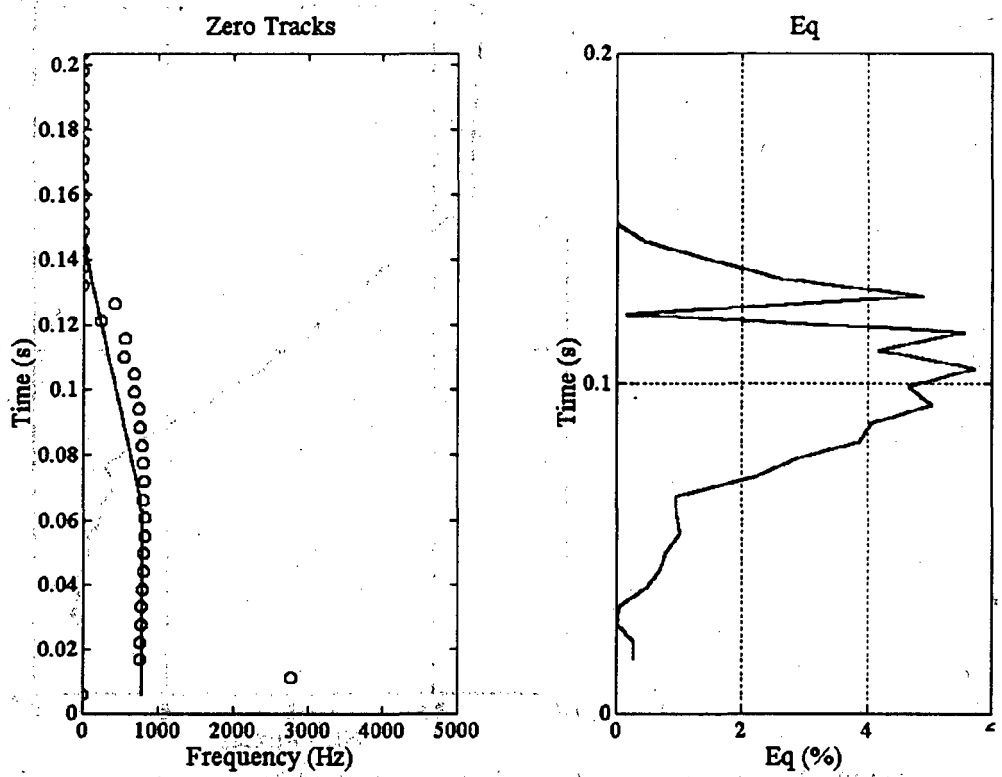


Figure E.5 Zero tracks of the synthetic speech signal /m/-/i/ (syn_mi) versus the estimated tracks (FWRLS).

Stellenbosch University https://scholar.sun.ac.za

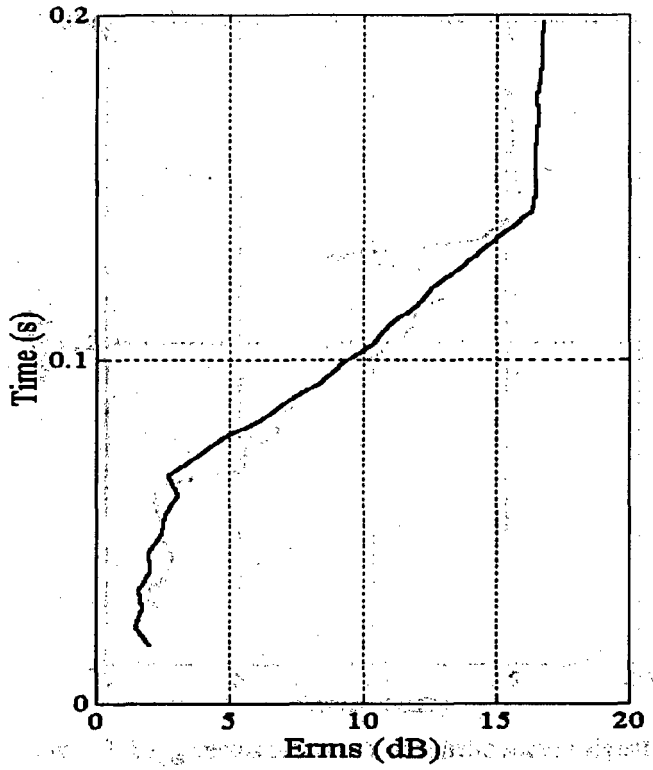


Figure E.6 E_{rms} versus time of the synthetic speech signal /m/-/i/ (syn_mi) (SEARMA).

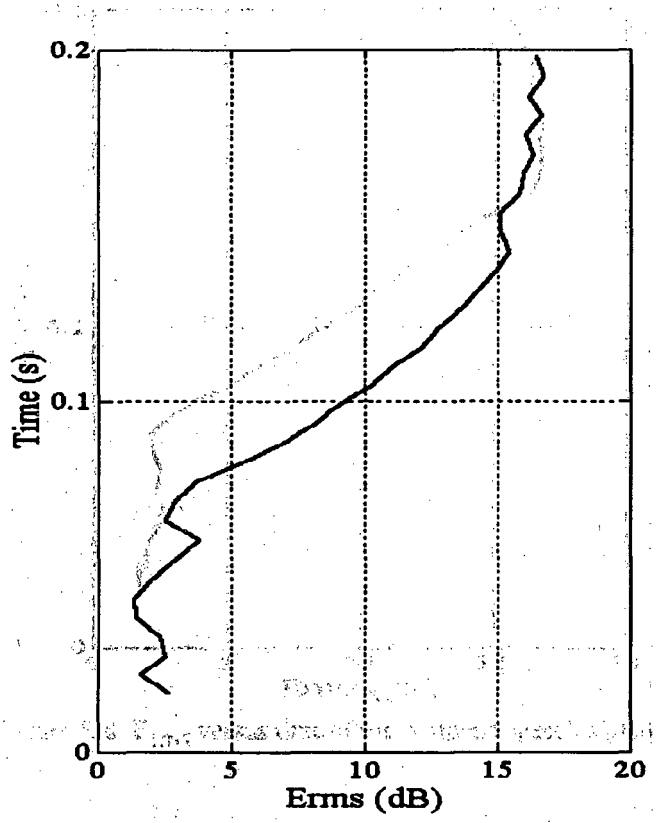


Figure E.6 E_{rms} versus time of the synthetic speech signal /m/-/i/ (syn_mi) (CWRLS).

Stellenbosch University https://scholar.sun.ac.za

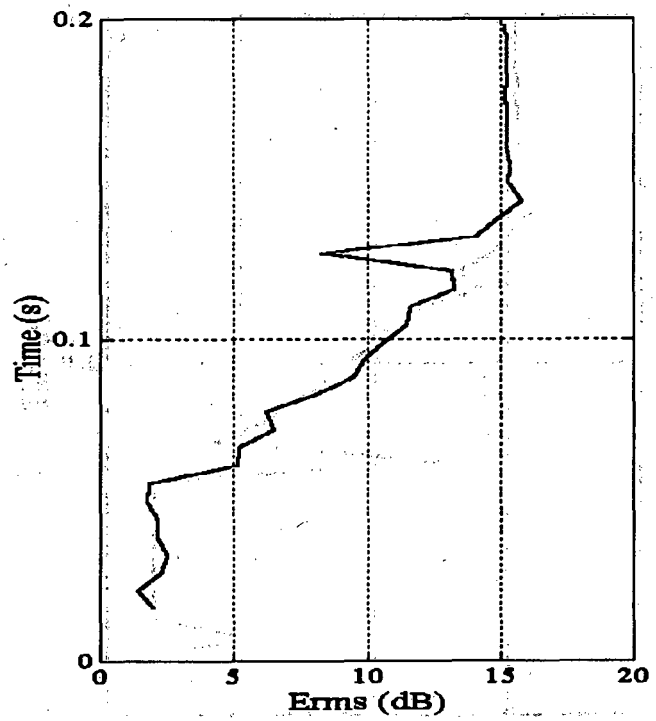


Figure E.7 E_{rms} versus time of the synthetic speech signal /m/-/i/ (syn_mi) (MWRLS).

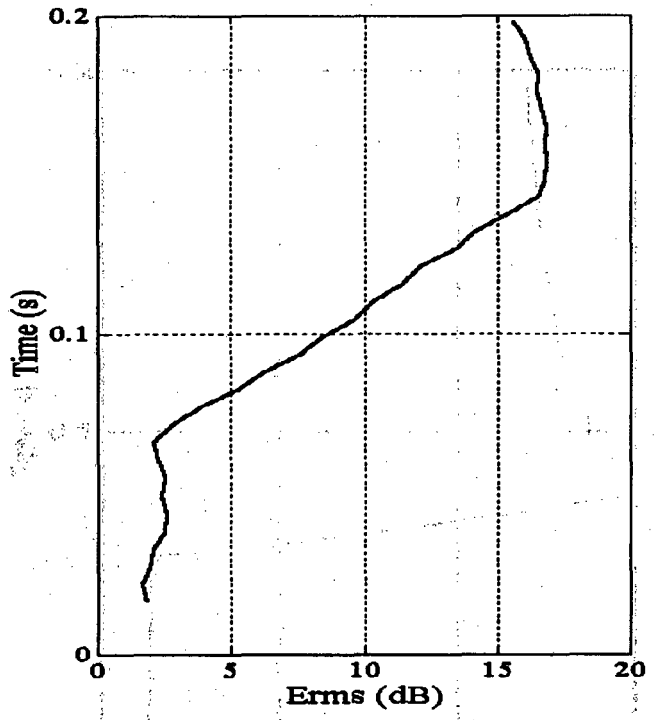


Figure E.8 E_{rms} versus time of the synthetic speech signal /m/-/i/ (syn_mi) (WRLS-VFF).

Stellenbosch University https://scholar.sun.ac.za

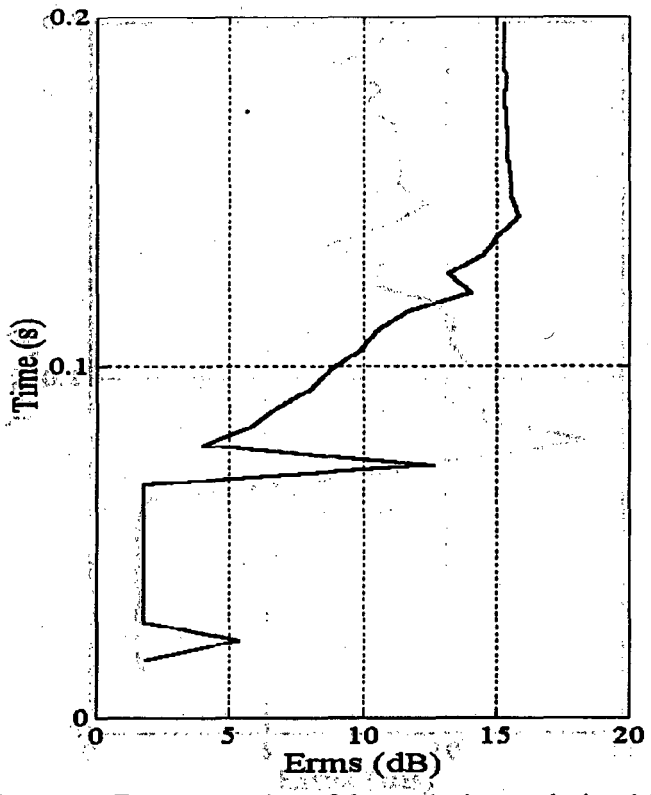


Figure E.9 E_{rms} versus time of the synthetic speech signal /m/-/i/ (syn_mi) (FWRLS).

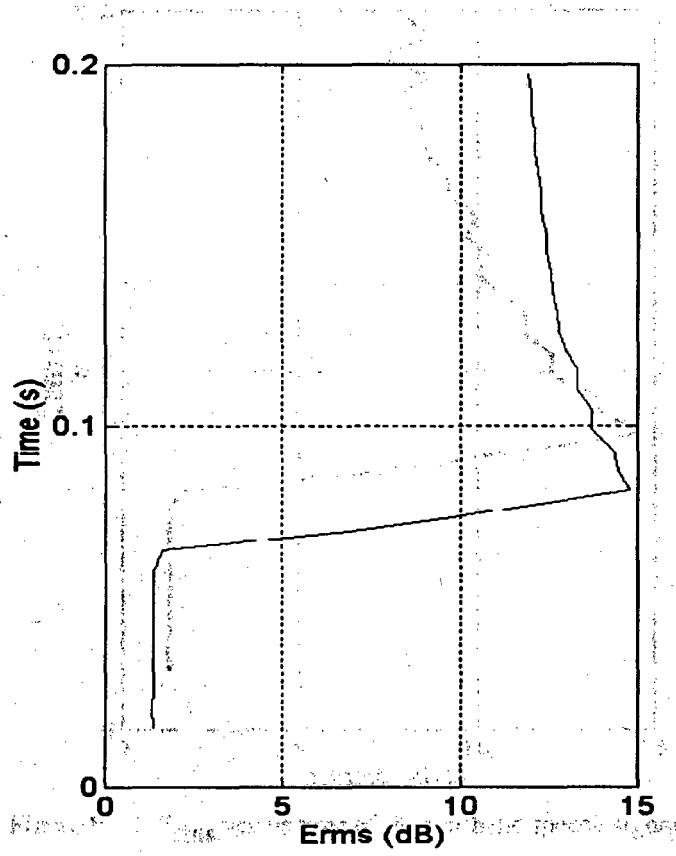


Figure E.10 E_{rms} versus time of the synthetic speech signal /a/-/i/ (syn_ai) (SEARMA).

Stellenbosch University https://scholar.sun.ac.za

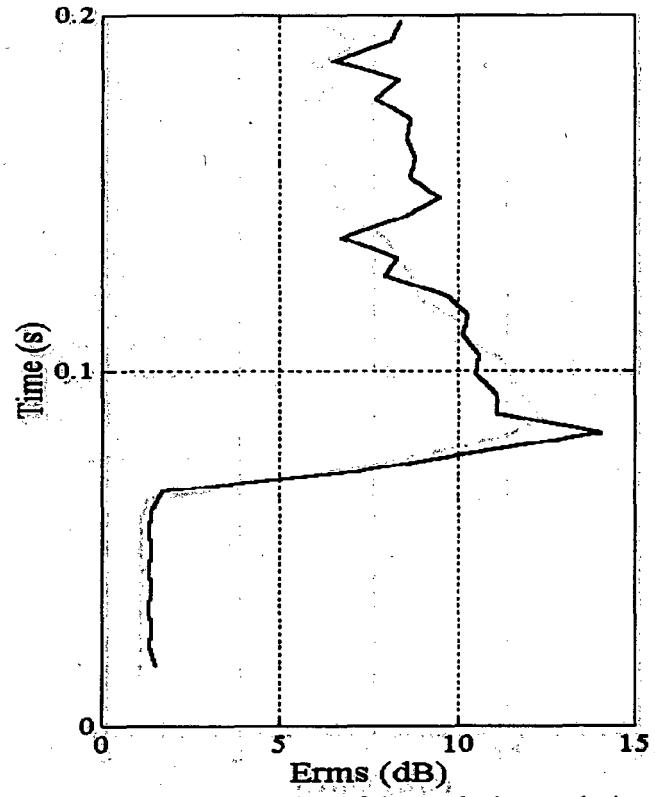


Figure E.11 E_{rms} versus time of the synthetic speech signal /a/-/i/ (syn_ai) (CWRLS).

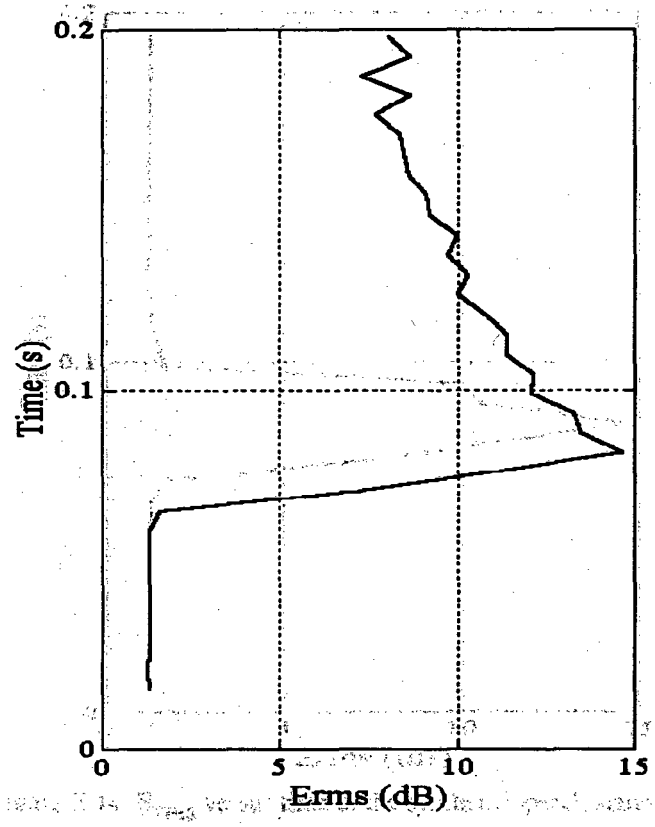


Figure E.12 E_{rms} versus time of the synthetic speech signal /a/-/i/ (syn_ai) (MWRLS).

Stellenbosch University https://scholar.sun.ac.za

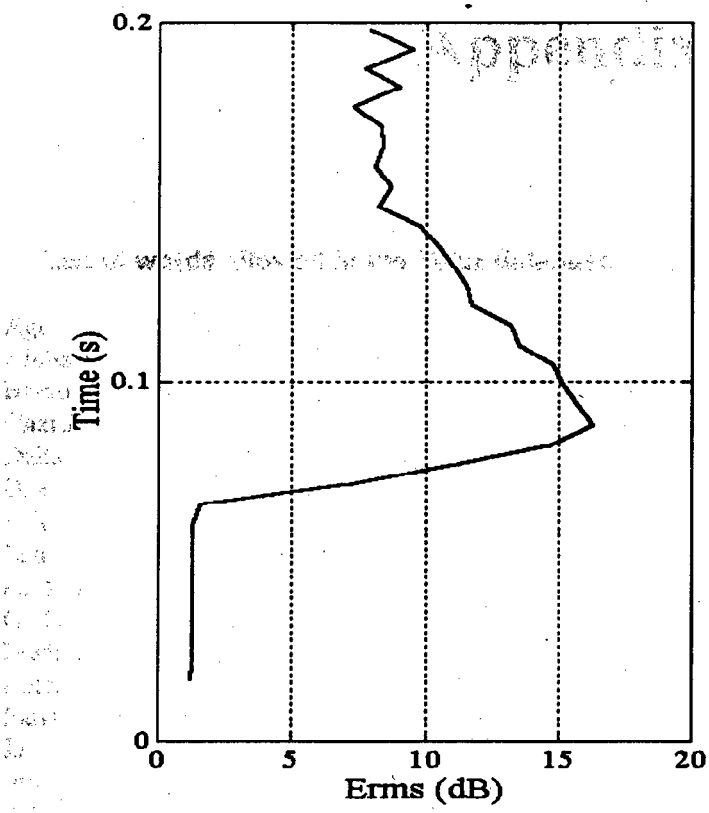


Figure E.13 E_{rms} versus time of the synthetic speech signal /a/-/i/ (syn_ai) (WRLS-VFF).

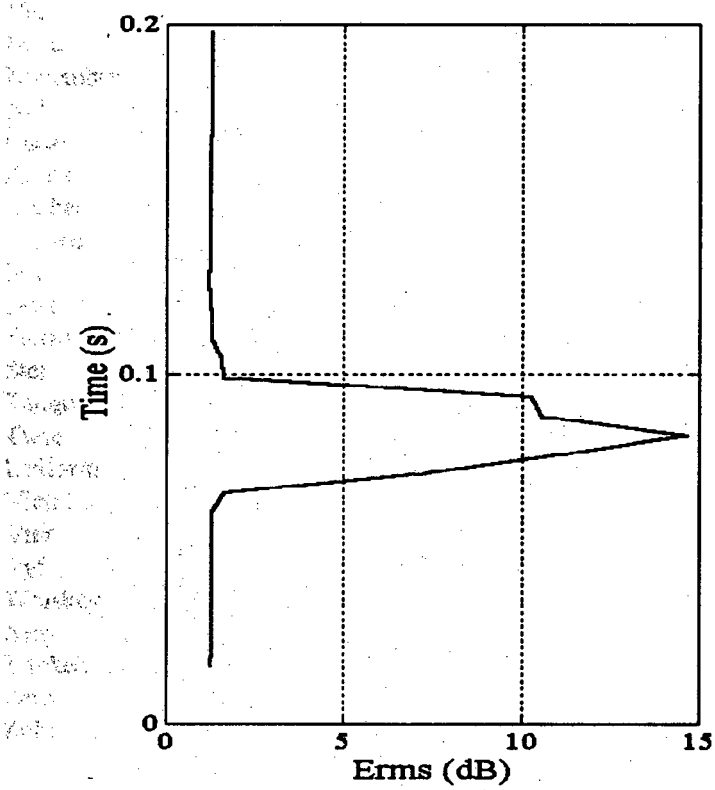


Figure E.14 E_{rms} versus time of the synthetic speech signal /a/-/i/ (syn_ai) (FWRLS).

Stellenbosch University https://scholar.sun.ac.za

Appendix F.

List of words allowed in the Telaz database.

Agt
Alpha
Bravo
Charlie
Delta
Drie
Echo
Een
Foxtrot
Golf
Hekkie
Hotel
India
Ja
Juliet
Kilo
Lima
Mike
Nee
Nege
November
Nul
Oscar
Pappa
Quebec
Romeo
Ses
Sewe
Sierra
Ster
Tango
Twee
Uniform
Victor
Vier
Vyf
Whiskey
Xray
Yankee
Zero
Zulu

Stellenbosch University https://scholar.sun.ac.za

Appendix G.

Mapping between the s-plane and the z-plane [53].

The complex variables s and z are related by:

$$z = e^{Ts}$$
$$= e^{T(\sigma + j\omega)}$$
$$= e^{T\sigma} e^{j(\omega T + 2\pi k)}$$

Where the bandwidth=$T\sigma$ and frequency=$\omega T + 2\pi k$ and $T = \frac{1}{f}$ is the sampling period.

k=0...∞.

For a specified bandwidth (B_1=60) and frequency (F_1=390) and sampling frequency (f=10000Hz) in the s-domain the z-plane representation is:

$$z_{F1/B1} = e^{\sigma T} e^{j(\omega T + 2\pi k)}$$
$$= e^{-(2\pi B1)\frac{1}{f}} e^{j\left(2\pi F1\frac{1}{f} + 2\pi k\right)}$$
$$= e^{-(2\pi 60)\frac{1}{10000}} e^{j(2\pi 390\frac{1}{10000})}$$
$$= .9930 e^{j0.2450}$$

Figure G.62 and G.63 shows the s-plane and z-plane representations respectively.

Stellenbosch University https://scholar.sun.ac.za

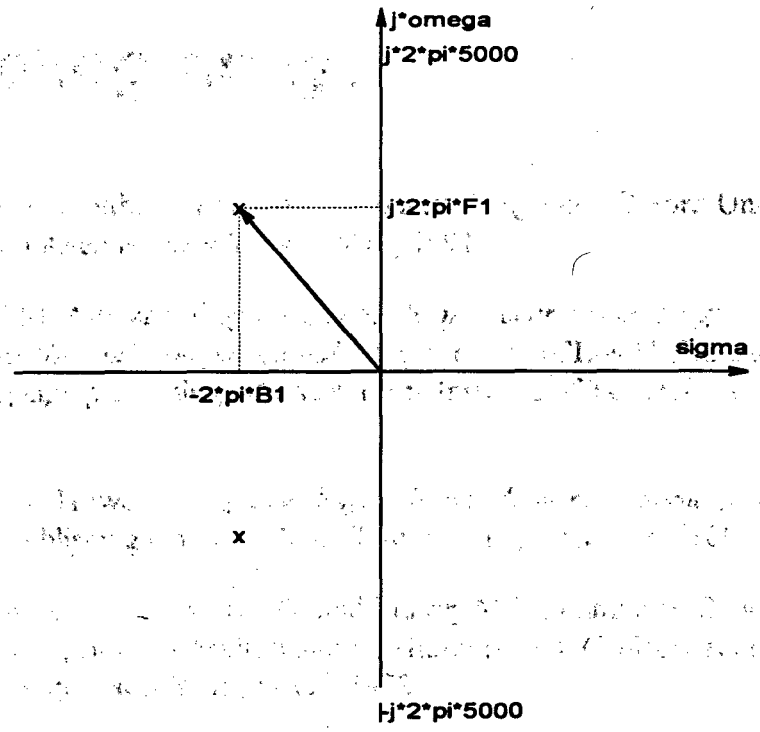


Figure G.62 S-plane representation.

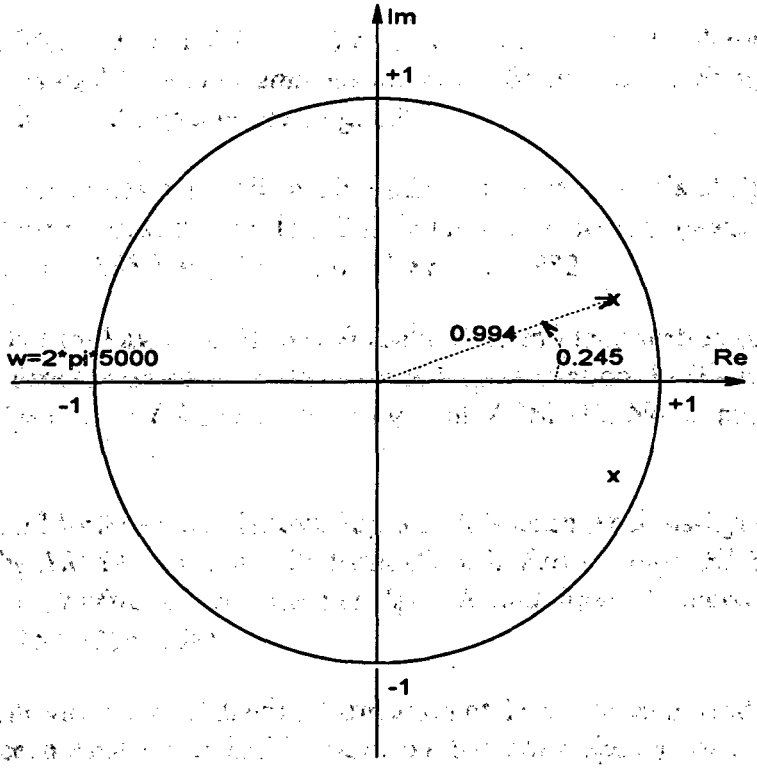


Figure G.63 Z-plane representation.

Stellenbosch University https://scholar.sun.ac.za

Bibliography

[1] George W. Smith. *Computers and Human Language.* Oxford University Press, 200 Madison Avenue, New York, 10016, 1991.

[2] Gerry T.M. Altmann. *Cognitive Models of Speech Processing: Psycholinguistic and Computational Perspectives.* ACL-MIT Press series in natural language processing, Massachusetts Institute of Technology, Cambridge, 1990.

[3] Darlene V. Howard. *Cognitive Psychology: Memory, Language, and Thought.* Macmillan Publishing Co., Inc., 866 Third Avenue, New York, 10022, 1983.

[4] Liberman A.M., Mattingly I.G. and Turvey M.T., Language Codes and Memory Codes, In A.W. Melton and E. Martin (Eds.). *Coding Processes in Human Memory.*, New York, Wiley, 1972.

[5] Shikano Kiyohiro. *Evaluation of LPC Spectral Matching Measures for Phonetic Unit Recognition.*, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA 15213; 3 Feb 1986.

[6] Van der Merwe C.J. and J.A. du Preez., *Calculation of LPC-based cepstrum coefficients using Mel-scale frequency warping.*, South African Symposium on communications and signal processing, 1991.

[7] Hiroyoshi Morikawa and Hiroya Fujisaki. Adaptive Analysis of Speech Based on a Pole-Zero Representation. In *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, No. 1, pp. 77-88, Feb 1982.

[8] Hiroyoshi Morikawa and Hiroya Fujisaki. System Identification of the Speech Production Process Based on a State-Space Representation. In *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, No. 2, pp. 252-262, Apr 1984.

[9] Hiroyoshi Morikawa and Hiroya Fujisaki. A Speech Analysis-Synthesis System Based on the ARMA Model and its Evaluation. In *Proceedings IEEE-IECE ASJ International Conference on Acoustics, Speech, and Signal Processing*, Tokyo, Japan, pp. 1253-1256, 1986.

[10] Hiroyoshi Morikawa. Adaptive Estimation of Time-Varying Model Order in the ARMA Speech Analysis. In *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-38, No. 7, pp. 1073-1083, Jul 1990.

Stellenbosch University https://scholar.sun.ac.za

[11] Yoshikazu Miyanaga ,Nobuhiro Miki, Nobuo Nagai and Kozo Hatori. A speech analysis algorithm which eliminates the influence of pitch using the model reference adaptive system. In *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, No. 1, pp. 88-96, Feb 1982.

[12] Yoshikazu Miyanaga and Nobuhiro Miki. Adaptive Identification of a Time-Varying ARMA Speech Model. In *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 3, pp. 432-433, Jun 1986.

[13] Erlendur Karlsson. RLS Polynimial Lattice Algorithms for Modelling Time-Varying Signals. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3233-36, 1991.

[14] Y.T. Ting and D.G. Childers. Speech Analysis Using the Weighted Recursive Least Squares Algorithm with a Variable Forgetting Factor. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, N.Mex , pp. 389-392, 1990.

[15] Nancy Hubing and Kyung Yoo. Exploiting Recursive Parameter Trajectories in Speech Analysis. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I-125-28, 1992.

[16] Hubing N.E. and S.T. Alexander. Statistical analysis of the soft constrained initialization of recursive least squares algorithms. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, N.Mex , 1990.

[17] Larry Marple. A New Autoregressive Spectrum Analysis Algorithm. In *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, N0. 4, Aug 1980.

[18] Marple S.L. Jr., High resolution autoregressive spectrum analysis usong noise power cancelation. *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp 345-348, 1978

[19] Raman K. Mehra. On the Identification of Variances and Adaptive Kalman Filtering. In *IEEE Trans. on Automatic Control*, Vol. AC-15, No. 2, pp. 175-83, Apr 1970.

[20] Raman K. Mehra. On-Line Identification of Linear Dynamic Systems with Applications to Kalman Filtering. In *IEEE Trans. on Automatic Control*, Vol. AC-16, No. 1, pp. 12-21, Feb 1971.

[21] Atal B.S., and S.L. Hanauer, Speech Analysis and Synthesis by Linear Prediction of the Speech Wave. In *J. Acoust. Soc. Am.*, vol. 50, pp 637-655, 1971.

Stellenbosch University https://scholar.sun.ac.za

[22] Atal B.S., Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. In *J. Acoust. Soc. Am.*, vol. 55, pp 1304-1312, 1974.

[23] Gold B., R.A. Rabiner., Analysis of digital and analog formant synthesizers. In *IEEE Trans. on Audio and Electroacoustics*, Vol. AU-16, No., pp. 81-94, 1968.

[24] Klatt D.H., Software for a cascade/parallel formant synthesizer. In *J. Acoust. Soc. Am.*, vol. 67, pp 971-995, Mar 1980.

[25] Pinto N.B. and Childers D.G., Formant speech synthesis: Improving production quality. In *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-37, N0. 12, Dec 1989.

[26] Lehmer, A Machine method for solving polynomial equations. *J. Assoc. Comput. Mach.*, vol. 8, pp151-162, 1961.

[27] Rissanen J., Modeling by shortest data description. *Automatica*, vol. 6, pp461-464, 1978.

[28] Akaike H. , A new look at the statistical model identification. In *IEEE Trans. on Automat. Contr.*, vol. AC-19, pp716-723, Dec 1974.

[29] Parzen E., Some recent advances in time series modeling. In *IEEE Trans. on Automat. Contr.*, vol. AC-19, pp723-730, Dec 1974.

[30] Rissanen J., Stochastic complexity and modeling. *Annals of Statistics.*, vol. 14, pp1080-1100, 1986.

[31] Hannan E.J., The estimation of the order of an ARMA process. *Annals of Statistics.* , vol. 8, pp1017-1081, 1980.

[32] Pukkila T.M., and P.R. Krishnaiah, On the use of autoregressive order determination criteria in univariant white noise tests. In *IEEE Trans. on Acoustics, Speech, and Signal Processing.* , ASSP-36, pp764-774, May 1988.

[33] Gavrishchuk V.V. and A.F. Starskov, A method for solving the Ricatti equation in filtering problems. *Sov. J. Comput. Syst. Sci.*(U.S.A.), vol. 30, No. 2, pp156-163, 1991.

[34] Lee D.T.L., M. Morf and B. Friedlander, Recursive least squares ladder estimation algorithms. In *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, N0. 3, Jun 1981.

[35] Marden, The geometry of zeros in a polynomial in a complex variable. *Amr. Math. Soc. Surveys*, New York, no.3, chap. 10, 1949.

Stellenbosch University https://scholar.sun.ac.za

[36] Juang B. H., L.R. Rabiner, On the use of bandpass liftering in speech recognition., In *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, N0. 7, Jul 1987.

[37] Rabiner L.R., B.H. Juang, Levinson S.E., Sondhi M.M, Recognition of isolated digits using hidden Markov models with continuous mixture densities., *AT&T Bell Lab. Tech. Journal*, Vol 64(6), Part 1, Aug 1985.

[38] Rabiner L.R., B.H. Juang, Levinson S.E., Sondhi M.M, Some properties of Continuous Hidden Markov Models Representations., *AT&T Bell Lab. Tech. Journal*, Vol 64(6), Part 1, Aug 1985.

[39] Rabiner L.R., B.H. Juang, An introduction to hidden Markov models., In *IEEE ASSP Magazine, pp 4-16*, Jan 1986.

[40] Schür, Über potenzreihen, die im innern des Einheitskreises beschränkt sind. *J. Reine Angew. Math.*, vol 147, pp. 205-232 (see also vol 148, pp. 122-145), 1917.

[41] Cohn, Über die Anzahl der Wurzeln einer algebraischen Gleichung in einem Kreise. *Math. Z.*, vol. 14, pp 110-148, 1922.

[42] T.J. Ulrych and R.W. Clayton, Time Series Modeling and Maximum Entropy, *Phys. Earth Planet. Inter.*, vol. 12, pp. 188-200, August 1976.

[43] A.H. Nutall, Spectral Analysis of a Univariate Process with Bad Data Points, via Maximum Entropy and Linear Predictive Techniques, *Naval Underwater Systems Center Technical Report TR-5303*, New London, Conn., March 1976.

[44] Hendrik F.V. Boshoff, A fast box counting algorithm for determining the fractal dimension of sampled continuous functions, *Proceedings IEEE COMSIG*, Cape Town, 11 Sept 1992.

[45] Burg J.P., *Maximum entropy spectral analysis*. Ph.D. Dissertation, Stanford University.

[46] Robert Lehmensiek, *"Segmentering van vloeiende spraak met fraktale metodes"*, Final year B.Eng. Project, Department of Electrical and Electronic Engineering, University of Stellenbosch, Stellenbosch, Nov 1992."

[47] M.W. Coetzer. *"'n Tweestapprosedure vir die ontleding en vergelyking van 'n aantal metodes vir die skatting van drywingsdigtheidspektra."*. PhD thesis, Department of Electrical and Electronic Engineering, University of Stellenbosch, Stellenbosch, Dec 1983.

[48] C.J. van der Merwe. *Phonetic Alignment of Speech using Continuous Density Hidden Markov Models.* , M.Eng. project, Department of Electrical and Electronic Engineering, University of Stellenbosch, Stellenbosch, Oct 1991.

Stellenbosch University https://scholar.sun.ac.za

[49] Douglas O'Shaughnessy. *Speech Communication: Human and Machine.* Addison-Wesley Series in Electrical Engineering: Digital Signal Processing, Massachusetts, USA, January, 1990.

[50] Lawrence Marple, Jr. *Digital Spectral Analysis with Applications.* Prentice-Hall International Inc., Englewood Cliffs, New Jersey, 07632, 1987.

[51] Steven M. Kay. *Modern Spectral Estimation: Theory & Application.* Prentice-Hall International Inc., Englewood Cliffs, New Jersey, 07632, 1970.

[52] Katsuhiko Ogata. *Modern Control Engineering.* Prentice-Hall International Inc., Englewood Cliffs, New Jersey, 07632, 1987.

[53] Katsuhiko Ogata. *Discrete-Time Control Systems.* Prentice-Hall International Inc., Englewood Cliffs, New Jersey, 07632, 1987.

[54] Peter V. O'Neil. *Advanced Engineering Mathematics.* Second Edition, Wadsworth Publishing Company, Belmont, California, 1987.

[55] Alan V. Oppenheim and Ronald W. Schafer. *Digital Signal Processing.* Prentice-Hall International Inc., Englewood Cliffs, New Jersey, 07632, 1975.

[56] Simon Haykin. *Adaptive Filter Theory.* Second Edition, Prentice-Hall International Inc., Englewood Cliffs, New Jersey, 07632.

[57] Rabiner E.L. and Schafer R.W., *Digital Processing of Speech Signals.*, Prentice-Hall International Inc., Englewood Cliffs, New Jersey, 07632, 1978.

[58] Murray R. Spiegel. *Mathematical Handbook of Formulas and Tables.* Shaum's Outline Series, McGraw-Hill Book Company, New York, 1968.

[59] John G. Proakis & Dimitris G. Manolakis, *Digital Signal Processing Principles, Algorithms and Applications.* Second Edition, Macmillan Publishing Company, U.S.A, 1992.

[60] John A. Borrie. *Modern Control Systems: A Manual of Design Methods.* Prentice-Hall International Inc., UK, 1986.

[61] Bozic. *Digital and Kalman Filtering.* Edward Arnold (publishers) Ltd, 41 Bedford Square, London, UK, 1979.

[62] Tretter, *Introduction to Discrete-Time Signal Processing.* Wiley, New York, 1976.

[63] Chris H. Pappas & William H Murray,III. *Borland C++ Handbook.* Second Edition, Osborne McGraw-Hill, 2600 Tenth Street, Berkeley, California 94710, U.S.A., 1992.

Stellenbosch University https://scholar.sun.ac.za

Index